An Agent-Based Framework for Artificial Stock Markets

Katalin Boer Mark Polman Arie de Bruin Uzay Kaymak

Erasmus University Rotterdam, Faculty of Economics
P.O. Box 1738, 3000 DR, Rotterdam, the Netherlands

Abstract

Stock markets strive to provide an efficient trading platform for investors. Trading rules and mechanisms issued to accomplish this differ among stock markets, and are subject to modification over time. Furthermore, market participants assume a broad range of roles and trading strategies. Such variation poses problems to those involved in the study of market dynamics, when developing an artificial stock market for experimentation and analysis. More than once, the resulting artificial stock markets, and thus the experimental results, are based on very restrictive assumptions. This paper introduces an agent-based framework for artificial stock market development and experimentation. The framework is flexible in the sense that multiple market structures are supported, and an infinite range of trading strategies by market participants can be captured. Such features are accomplished through the configuration of framework properties, and the appropriate hooks for extension of the framework's components.

1 Introduction

Artificial stock markets are models of real stock markets, designed with the aim to study market dynamics. In agent-based artificial stock markets software agents are used to represent traders on a market. The idea to use intelligent software agents for studying market dynamics originates from the agent-based computational economics approach, the computational study of economies modelled as evolving systems of autonomous interacting agents[13].

The motivation for the study of market dynamics, which are generally poorly understood, is manifold. Investors aspire profit increase, market regulators improved market quality, and scientists deeper insight in the workings of financial theories and hypotheses.

What makes market dynamics difficult to understand is the dynamic, complex feature of price formation process, which, to make things more complicated, differs among markets, and changes within markets over time. It is governed by a market's trading rules, the roles adopted by market participants and their (hidden) trading strategies.

Given these characteristics of markets and traders it is difficult, if not impossible, to design a market that perfectly reflects all the details of a real stock market. Therefore several choices, simplifications and assumptions are needed in order to make attempts to represent market structures and traders' behavior. Given the very specific market structures of the artificial stock markets (ASM) in the literature, findings might hold only in the specific settings and might not globally reflect market dynamics. The way ASM's are designed lacks the flexibility required to analyze the impact of the restrictions made.

This paper introduces an agent-based framework for artificial stock market development and experimentation. Through proper configuration, multiple market structures are supported. Furthermore, by extending selected framework components, an infinite range of trading strategies by market participants can be captured. This allows for the analysis of market dynamics on many market types, and for rapid evaluation of alternatives.

The outline of this paper is as follows. Section 2 assesses software agents' adequacy for studying market dynamics and briefly discusses how agent-based ASM's are designed in the literature. Section 3 presents the agent-based framework that we are designing, whereas Section 4 describes how one can use it to develop artificial markets. Final discussions and future directions are provided in Section 5.

2 Agent-based artificial stock markets

Agent-based computational economics (ACE) studies how global regularities arise from individual interactions[13]. Here, individuals are represented by (software) agents interacting in an artificial environment. By using agents in studying market dynamics, heterogeneous, boundedly-rational, and adaptive behavior of individuals can be represented and its impact on market dynamics assessed.

Market dynamics are studied through price dynamics (usually in form of returns). Price formation on a market depends on the market structure . The market structure is defined by trading rules and trading systems and determines, for example, the role, tasks, rights of different participants, the assets that can be traded, trading times and the information that is available [6].

For example, on a so-called *call-market*, traders can place their orders at well-defined points in time, and the new market price is based on aggregated orders. On *continuous markets* orders may be placed at any time the market is open, and prices arise whenever an order can be executed. Combinations of continuous sessions with call sessions are also very common[4].

Apart from price formation, the market structure also determines many of the trading actions by market participants. Based on their tasks and role in the market we distinguish three types of participants (traders): *investors*, *brokers* and *market makers*[2]. Investors (e.g. individuals, or money managers) are simple traders, and generally not considered part of the market itself. Brokers are primarily required to execute orders for investors. Market-makers (e.g. the "Specialist" on the NYSE, or the "Hoekman" on he Amsterdam Stock Exchange) are responsible for the liquidity of the stocks assigned to them, and generally have to provide *bid* *and ask quotes*. The quotes must specify the number of shares and the price for which market makers are prepared to buy and sell. Traders maintain a so-called *order-book* to store unexecuted orders[4].

Traders differ in the strategies adopted to place orders, execute orders, provide liquidity, determine bid and ask quotes, etc. Their decisions are influenced by several factors, besides the market structure , such as: belief, capabilities, news, economy, preferences, and financial situation. Traders generally do not reveal their strategies, which makes market dynamics difficult to understand.

In existing literature, much of the above mentioned variability is abstracted from. Despite the fact that most of the stock markets apply continuous trading a very common choice is the representation of call markets. In [7], [10] and [8] for example orders are aggregated and matched at single price and all the traders (or randomly selected groups of traders) simultaneously place orders. An attempt at a continuous trading model is made in [12] where traders do not make constant decisions but they "sleep" after actions and "wake up" at predefined times, or as a result of events. Other studies model continuity by randomly [11] or stochastically [9] selecting one trader whose decision is carried out, and automatically matching new orders with pending ones if possible.

The problem with centrally selecting agents whose actions will be carried out, is that, in this way, the decisions of some traders are not taken into account (e.g. in [9]). Thus, agents are no longer autonomous regarding their actions. Autonomous behavior can only be accomplished if the agents themselves decide when they want to trade as is the case in real markets.

As far as trader modelling is concerned, most ASM's focus on representing investor behavior. There are no artificial stock markets to our knowledge that include brokers, and only a few markets with market makers (e.g. [3]), however their behavior directly affects price formation. Investors are differentiated in literature by their belief (expectation) regarding future values: fundamentalists believe that the price will achieve its real fundamental value, while technical analysts try to find patterns in historical data (e.g. [7, 8, 9]). Traders in artificial stock markets usually determine their orders based on a function that maximizes their utility accompanied by a stochastic value or stochastic function (e.g. in [8, 9]). Further, investors who trade randomly or according to some distribution function are seen as well (e.g. [12, 11]). Besides the strategies implemented in the ASM literature several others are described by empirical and behavioral studies, and many more exist in reality.

The vast number of trading strategies in a broad range of market structures motivated us to design a framework that accommodates this diversity. This framework is the topic of the remainder of this paper.

3 On the design of the framework

This section introduces an agent-based framework for the development of the artificial stock markets. The basic conceptual design regarding the architecture of traders is described in [2]. The idea behind that specific primary architecture is to focus on rarely implemented, continuous market organizations and asynchronously interacting role-based agents. Recent improvements primarily concern increased flexibility: the new framework can be configured to experiment with various market types, and can be extended to experiment with different trading strategies. It includes a variable market structure and skeletons for the three trader types: market makers, brokers and investors. The skeletons provide the basic structure and generic behavior of these trader types. They are implemented as autonomous software agents concurrently carrying out multiple tasks. In the new framework traders' behaviors (strategies) can be implemented on top of the skeletons. In this section we emphasize the difference between the generic behavior of agents within a role which is included in skeletons, and the variable tasks that can be attached to the skeletons.

3.1 The market structure

The design of the market structure is based on the institutional organization of stock markets. Given that most of the real stock markets allow for continuous trading, the primary focus during the design of our framework has been to support trading continuity. We allow for two types of continuous market structure s: with and without continuous double-auctions. A market with continuous double-auction represents for example the trading mechanism of the NYSE.

However, since most artificial markets implement call-auctions, our framework also allows for markets where trading occurs at discrete points in time. In addition it provides the possibility to implement hybrid markets (e.g. continuous with call-auctions to determine the opening and the closing prices) given that the organization of markets in this way is very common[4].

In order to change the continuous market structure into a call-auction or hybrid market structure we have to simplify its behavior. This is achieved by having a specific agent that notifies traders whenever a new call-auction starts, and making traders react to such notifications in an appropriate manner.

In contrast to existing artificial stock markets, featuring only investors and market makers, the introduced framework includes a third trader group: the brokers. However, if necessary they can be excluded, for example, in order to replicate experiments of ASM's that do not use negotiating brokers.

3.2 The skeleton of the investors

From the viewpoint of an outsider, an investor collects information and places orders. Accordingly, the skeleton of an investor-agent performs the following generic behavior: it continuously receives and analyzes messages, places orders according to a given trading strategy and sends orders to an intermediary (broker or market maker). Investors can search for available intermediaries in a "yellow page", that contains the name of agents on the market, together with the services that they provide.

The trading strategy applied by investors depends on several factors, such as their expectations, beliefs and intention, and results in orders. The chosen strategy for solving this complex decision problem is what distinguishes investors from one another. The skeleton does not provide a specific strategy, but an empty placeholder on top of which different strategies can be implemented.

3.3 The skeleton of the brokers

The brokers' primary task is to execute orders on behalf of the investors. Accordingly, the skeleton of the broker-agents takes care that brokers continuously receive orders from investors and try to execute these at an improved price.

The main decision problem that brokers face is which of the received orders to execute next, and how to execute it. There are several possibilities for brokers to select orders for execution. One option is to select them one by one, for example in the order of arrival. Another is to consider current market prices and execute the orders with the highest probability for execution. A third option is to somehow aggregate a set of orders and create a new order that contains and represents them.

Additionally, brokers have to decide how to execute the selected order. Depending on the market structure they can decide to take the other side of a received order, to negotiate it for an improved price or to route it to the market maker for further execution. If negotiation (double-auction) is allowed on the market, and brokers choose to negotiate, they also need to adopt a negotiation strategy. Such a strategy involves, for example, the determination of the time-length during which they are trying to negotiate and the values with which negotiation prices should change.

It is not clear how in reality brokers solve all the decision problems they face. Allowing for multiple strategies enables us to study how the brokers' success and the market dynamics depend on the strategy applied. Hence, the skeleton, again, provides only the implementation of the generic behavior of broker-agents without a concrete strategy-implementation, which the user needs to provide.

3.4 The skeleton of the market makers

The third skeleton provides the basic functionality of a market maker by implementing the generic behavior of market makers. It lets market makers continuously receive orders from other market participants. When an order is received the (skeleton of the) market maker agents attempts to execute it against its quoted bid/ask values (they are obliged to do this in reality). If complete or partial execution of the order is possible the transaction is conducted, reported, and confirmed to the requestor.

If the order can be executed only partially, the market maker has to decide whether to execute the rest (and if so, how to execute it: for example with other orders from its order book) or to enter it into the order-book as well. After almost any order-reception or execution new bid and ask quotes are issued. Bid and ask quotes can also be updated if the market maker decides that there is not enough activity regarding a certain stock.

Market makers apply various strategies to set their bid and ask quotes, and to decide what to do with orders that do not match their quoted bid and ask quotes.

In order to allow for experiments with multiple market maker types, the skeleton again does not provide a concrete solution to these decision problems, but empty placeholders to be extended with user-defined strategies.

4 Implementing ASM's on top of the framework

In order to develop a working market on top of the framework described in the previous section, the market structure has to be specified and strategies for the decision problems of the traders have to be implemented. Two kind of information should be provided a-priori: market organization specific and trader specific.

4.1 Market specific configurations

The type of the market (call, continuous or hybrid) can be given in configuration files. Depending on the chosen structure, additional specifications should be provided, such as: the time-period during which traders may place orders after an auction starts; indications whether the opening/closing price of a hybrid market is defined by a call-auction; whether call-auctions should be launched in a hybrid market when extreme price changes occur; the number of call-auctions during a trading day if the market is characterized by a pure call-auction mechanism; whether brokers are allowed to negotiate within the market maker's quote in case of call-auctions.

4.2 Trader specific configurations

The generic behavior of traders within a role is implemented by the software agents included in the skeletons. However, in order to make traders actually function and display different behaviors, the placeholders in the skeletons, as outlined in the previous section, need to be filled with actual strategies (i.e. approaches for solving the various decision problems). For example, for a newly introduced market maker a bid/ask setting strategy is required, whereas an investor needs an order placing strategy, and a broker a negotiation strategy.

Strategies adopted to solve the same decision problem can differ in their number and type of parameters. For example, the strategy of a market maker that defines bid/ask spread based only on the content of the order book, differs from the one that sets it as described by the Glosten-Milgrom model (like in [3]), where additional parameters are required (e.g. upper and lower limits of the fundamental value). All this, however, does not need to be hard-coded in the skeletons. Strategy-attributes can be provided to the skeletons in simple structured files. Skeletons need only to know the names of these files and the name of the correspondent strategy.

4.3 Launching the application

In order to "open" a market, one needs to run a main application with the specified market structure, the market makers and eventually the brokers. In order to let trading on the market begin, one needs to start investors with implemented trading strategies. Multiple investor-agents can join the market from different computers, any time during market operation. The only information they require is the name of the computer running the market.

4.4 Technical details

The framework has been implemented using JADE (Java Agent DEvelopment Framework)[1]. We implemented basic investor, broker, and market maker behavior as agents using the generic agent features offered by JADE. The behavior model of JADE provides for the execution of concurrent agent activities. Continuity is modelled by concurrent execution of agent actions (in Java threads) who interact by asynchronous message passing.

In order to let strategies vary among traders, we apply the "strategy behavioral pattern" described in [5]. User-defined strategies are encapsulated in strategy classes, extending an abstract strategy interface (this is the aforementioned empty placeholder in the skeletons of Section 3). Strategies and their attributes have to be declared and described in structured files (XML-files). At market startup, the framework will initialize trader agents according to the appropriate configuration files, and assign them the required strategies. From then on, the thread-based execution mechanism of JADE lets the agents continue their programmed behaviors.

5 Discussion and future research

In this paper we have presented an agent-based framework supporting the most typical market structures (including continuous trading sessions) and basic trader roles. We have briefly described how various types of markets can be configured and how multiple types of traders can be implemented using this framework.

The primary aim of this framework is to provide flexibility in modelling markets and traders' behavior and thus achieve a more detailed representation of stock markets than current studies. It should however, be analyzed in which way such detailed representation affects market dynamics.

Through its flexibility the framework improves the possibility to validate existing ASM's. It provides a testing bed for replicating the experiments of existing artificial stock markets and for analyzing their findings. Its variable market structure enables us to examine whether findings in literature hold only in their specific settings or are generally applicable.

The description given in this paper concerns the design and primary implementation phases of the framework within an ongoing research. After testing it sufficiently, we aim to make it publicly available so that additional experiments can be conducted by the research community.

References

[1] Fabio Bellifemine, Giovanni Caire, T. Trucco, and G. Rimassa. Jade programmer's guide. 2002.

[2] Katalin Boer and Uzay Kaymak. Microsimulation of artificial stock markets based on trader roles. In *International Workshop on Data Mining and Adaptive Modelling Methods for Economics and Management (IWAMEM-03)*, pages 61–72, 15-16 September, Porto, Portugal 2003.

[3] Sanmay Das. An agent-based model of dealership markets. In *International Workshop on Complex Agent-based Dynamic Networks, Oxford,*, October 2003.

[4] Marianne Demarchi and Thierry Foucault. Equity trading systems in europe: A survey of recent changes. *Annales D'Economie et de Statistique*, (60):73–115, October-December 2000.

[5] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns. Elements of Reusable Object-Oriented Software.* Addison-Wesley Longman, Inc, 1997.

[6] Larry Harris. *Trading and Exchanges: Market Microstructure for Practitioners.* Oxford University Press, 2003.

[7] Cars H. Hommes. Financial markets as nonlinear adaptive evolutionary systems. *Quantitative Finance*, 1:149–167, 2001.

[8] Blake LeBaron. Building the Santa Fe artificial stock market. *Working Paper, Brandeis University*, 2002.

[9] O. Loistl and O. Vetter. *KapSyn Computer Modelled Stock Exchanges, User Manual. Ver. 3.02.*, 2000.

[10] L. Matassini and F. Franci. On financial markets trading. *Physica A*, (289):526–542, 2001.

[11] M. Raberto, S. Cincotti, S.M. Focardi, and M. Marchesi. Agent-based simulation of a financial market. *Physica A*, (299):319–327, 2001.

[12] M. Shatner, L. Mushnik, M. Leshno, and S. Solomon. A continuous time asynchronous model of the stock market. *arXiv:cond-mat*, 2000.

[13] Leigh Tesfatsion. Introduction to the special issue on agent-based computational economics. *Journal of Economic Dynamics and Control*, 25:281–293, 2001.