KEY POINTS FOR REALISTIC AGENT-BASED FINANCIAL MARKET SIMULATIONS

Iryna Veryzhenko, Philippe Mathieu
LIFL, UMR CNRS-USTL 8022, Lille, France
iryna.veryzhenko@univ-lille1.fr, philippe.mathieu@lifl.fr

Olivier Brandouy
Sorbonne Graduate Business School, Paris France
olivier.brandouy@univ-paris1.fr

Keywords: Multi-agent systems, Artificial market, Market microstructure, Agents interactions

Abstract: The purpose of this paper is to define software engineering abstractions that provide a generic framework for stock market simulations. We demonstrate a series of key points and principles that has governed the development of an Agent-Based financial market in the form of an API. The simulator architecture is presented. During artificial market construction we have faced the whole variety of agent-based modeling issues and solved them : local interaction, distributed knowledge and resources, heterogeneous environments, agents autonomy, artificial intelligence, speech acts, discrete scheduling and simulation. Our study demonstrates that the choices made for agent-based modeling in this context deeply impact the resulting market dynamics and proposes a series of advances regarding the main limits the existing platforms actually meet.

1 INTRODUCTION

Multi-agent modeling is actively applied to financial market simulation, due to its possibility to reflect all complexities and automation of financial markets. Agent-based computational simulations (Woolridge, 2002) may contribute to the advances in finance. Examples of multi-agent and machine learning tools, applied in financial topics, are Bayesian learning, that can be used by agents to incorporate all available information into the decision making process (Mitchell, 1997), and techniques for tracking a moving parameter (Cesa-Bianchi and Lugosi, 2001), that are useful in estimating the possibly changing fundamental value of a stock. On the other hand, financial market is an important implementation area of agent-based modeling and machine learning, since agent objectives and interactions are clearly defined. For this reason, financial market environment can help to answer some modeling issues related to agent engineering or test robustness of existing behavioral models.

At the present there are large number of various agent-based frameworks, with varying functionality and architecture, addressing different problems. There are two major approaches to agent-based financial market simulations. The first one is realization of specific market type, with special agents behavior, trading instruments and rules. The other way of simulator realization is to design the general environment with flexible settings and functionality, that can accept heterogeneous agents populations. We list just some of such models, that bellow to these two approaches, and that point out some design questions. Altreva Adaptive Modeler is a tool for creating agent-based market simulation models for a specific problem : stock price forecasting (Witkam, 2003). The author, among other questions, describes a problem of a memory limitation during the framework processing. JLM market simulator is another tool that investors can use to create a market model using their own inputs. The authors (Jacobs Levy Equity Management, Inc.) conclude that it is not an easy task to build a complex asynchronous simulation with reasonably realistic properties. The first complexity is a diversity of agents behavior (for instance, there are only mean-variance investors in JLM), the other one is specification of user's trading strategies, that does not require the user's programming skills (Jacobs et al., 2004). Ascape is a general agent-based framework, developed at Brookings Institute in the 90s (Inchiosa and Parker, 2002), that is actively used in financial market modeling. Its' developers discuss a design possibilities to express the same basic modeling ideas in one way and have them tested in many different environments and configurations. Most of present artificial market platforms suffer from a lack of flexibility and must be viewed as software rather than as APIs, in a mean that they are oriented for specific problem solving, but cannot be used to explore a wide range of financial issues due to some structural choices made by the developer during the coding phase.

In this paper we present the ArTificial Open Market API (here-after ATOM, see `http://atom.univ-lille1.fr`) and focus attention on the important issues of agent-based stock market design. Among others, we make a specific point on the ability of this new, generic architecture to overcome some issues mentioned previously. During the stock market simulation we model individual market entities (agents, banks, participants), interactions between them, the behaviors and the decision-making process, global constraints in the interaction process, environment rules and external infrastructure. This framework allows to run large scale experiments under a variety of scenarios and conditions, to describe not only what have happened in the market or the generated price patterns, but also to investigate in fine grain the reasons through which interactions, strategies and rules, made it happen. Our financial market model is close to Influence Reaction Model for Simulation (IRM4S) (Michel, 2007), (Ferber and Muller, 1996), moreover, we show how existing models of the interactions between system entities (markets infrastructure, agents, banks) fit *Interaction Movement Computation* (*MIC**) abstract autonomous agents model (Gouaich et al., 2005). In our modeling, we move in the direction of a growing complexity, in the mean of agents intelligence (from zero-intelligent to cognitive traders) and their interactions (from individual independent behavior to the competitive interactions). We run series of validation tests in order to show the ability of ATOM to replicate real market features.

2 The Artificial Trading Open Market API

ATOM is an environment for Agent-Based simulations of stock markets. At the present moment, it is realized based on the architecture close to the Euronext-NYSE Stock Exchange one. Agent-Based artificial stock markets aim at matching orders sent by virtual traders to fix quotation prices. Price formation is ruled by a negotiation system between sellers and buyers based on an asynchronous, double auction mechanism structured in an order book (see Figure 1). ATOM is developed as large scale experimental

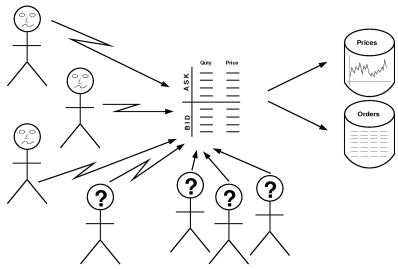


Figure 1: Double auction market with human and artificial traders

platform with heterogeneous agents populations. It is concrete implementation of MAS abstract design issues: *agent autonomy* and its *behavior* (strategy), each agent asses her position and makes decisions individually; price history and orders sets are *emergent phenomenon* of market activities; allowance of *distributed simulations* with many computers interacting through a network as well as local-host, extremely fast simulations; possibility to design experiments mixing *human beings and artificial traders*.

Distinctive features

Five distinctive aspects of ATOM can be highlighted:

1. ATOM can use various kind of sophisticated agents with their own behaviors and intelligence (see section 4), including portfolio optimizers as this platform provides the multi-assets (multi-orderbook) organisation. In addition, ATOM contains all variety of the Euronext-Nyse orders: limit, market, stop-limit,iceberg, etc.
2. Thousands of agents can evolve simultaneously, creating a truly heterogeneous population. Once designed, agents evolve by themselves, learning and adapting to their (financial) environment. 100,000 agents have been employed simultaneously for technical analysis evaluation research (Brandouy and Mathieu, 2007).
3. ATOM can combine human-beings and artificial traders on a single market using its network capabilities. This feature support a wide variety of configurations: from "experimental finance" classrooms with students, to competing strategies run independently and distantly by several banks or research labs. The scheduler can be set so to allow human agents to freeze the market during

their decision process or not (see above, section 3.4).

4. ATOM serves as a "replay-engine", meaning that it is possible to re-execute whole trading log file with following information: identification of order book, that corresponds to stock identity, agents identification, the platform time stamp fixed at the very moment orders arrive to a given order-book, prices resulting from the orders. ATOM takes less than 5 seconds to replay an entire day of trading, that contains 400,000 activities. This tool is extremely important for policy-oriented experiments focusing on the technical features of the market microstructure (tick size, price fixing protocol) and its influence on the price dynamic.
5. Agents can be viewed as simple nutshells in certain cases: they only execute actions predefine by third party, meaning that their behavior is defined (controlled) by experimenter. These agents are called "Hollow Agents". For example, a human trader can act through such agents. By definition, "Hollow Agents" do not have any artificial intelligence and can be assimilated to human-machine wrappers.

3 Artificial market design issues

Artificial Stock Market is environment to express all classical notions used in multi-agent systems. First of all, the environment in which agents evolve, as well as their behaviors and own dynamics, communication or interactions. ASM, like any other MAS, are suited for the study of various emergents phenomena. Using the so-called "vowels" approach (Ricordel and Demazeau, 2001), the definition of `AEIO` (`A` Agents, `E` Environment, `I` Interactions, `O` Organization) is straightforward. Nevertheless, if one wants to build an efficient platform, several issues can be identified and must be precisely and strictly regulated.

3.1 Entities organization

During the system modeling we employed many MAS design principles, for example, modularity and encapsulation, that suggest dividing the system into different sub-organizations, with the agents involved in the organization parts. The architecture can be viewed as a system with interacting components: i) *Agents*, and their behaviors, ii) *Markets* is defined in terms of microstructure and iii) *Bank* reflects intermediaries and monetary financial institutions iv) the *Artificial Economic World* provides economical indicators. Depending upon the researcher targets, the *Artificial Economic World* can be plugged or not in the simulations.

Based on the Gaia methodology (Zambonelli et al., 2003) for MAS design, we link each system entity with the sets of *Roles*. Thus, *Market* is responsible for the generating of market scenarios and price path, it presents set of constraints, rules, regulations, leading participants to activities. *Agents* may play roles of buyer or seller with different trading objectives. The agents directly initialize transactions. *Bank* component represents all intermediaries, that maintain information exchange between buyers and sellers. At the same time, Bank can be considered as the special type of buyer or seller, that has unlimited wealth, hence take active part in stock trading. We propose to consider Bank as trading and intermediary agent. *Artificial Economic World* provides external information about perspective corporates development, dividends and coupons changes, tax police modification. This information influences agent decisions. Artificial market architecture (system elements and interaction between them) is presented on the Figure 2. Thanks to its high modularity and its

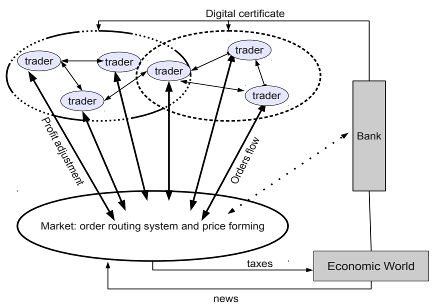


Figure 2: Market organizations and interactions

ability to mimic real-world environments, it can also serve as a research tool in Portfolio Management, Algorithmic Trading or Risk Management among others.
From a pure technological point of view, ATOM can also be viewed as an order-flow replay engine. It means that bankers can test their algorithmic-trading strategies using historical data without modifying the existing price series or backtest the impact of their trading-agents in totally new price motions or market regimes generated by artificial traders.

It is hardly possible to describe the complex algorithmic structures that are necessary for the realization of such multi-agent platforms; therefore we have chosen to introduce three of difficulties one must face while developing an ASM : i) the management of orders' ID, ii) the scheduling system, and iii) the introduction of a human-being in the simulation loop (here-after *"human-in-the-loop"* problem).

3.2 A unique identity for Orders

In its simplest form, an order is a triplet of direction (purchase or sale), a quantity and a price. Usually this type of order is called a *"Limit Order"*. In the Euronext-NYSE system, several other orders are used (see *"EURONEXT" Rule Book* at `http://www.euronext.com`). Once constructed by an agent, the order is sent to the order-book. It is then ranked in the corresponding auction-queue ("Bid" or "Ask" if it is an order to "Buy", respectively to "Sell") where are stacked the other pending orders using a "price-then-time" priority rule. As soon as two pending orders can be matched, they are processed as a "deal", which delivers a new price. Notice that the clearing mechanism implies that cash is transferred from the buyer to the seller and stocks from the seller to the buyer.

For various reasons, financial institutions need to proceed again an historical record of orders (for example, for the optimization of algorithmic trading methods). Such historical records collect the expression of human behaviors in specific circumstances. The first difficulty of order-flow replaying is exact interpretation of the order flow as it is expressed in the real-world. If order set consists of only *"Limit Orders"*, it could be perfectly reproducible. Unfortunately, issued orders can be modified or deleted. It implies one must be able to identify clearly reference between an "Update" or a "Delete" and previously issued orders. Thus, a generic platform has to use a unique ID for orders. This is particularly important in situations when several possible identification keys for orders potentially coexists : in the replay-engine situation (real-world orders ID and time stamps), in the Agent-Based platform mixing human beings and artificial traders situation (orders ID, platform time-stamp) or in any combination of these states. To our knowledge, this is neither the case for the Genoa Artificial Stock Market (see (Raberto et al., 2003)) nor in the Santa-Fe ASM for example (see (Le Baron, 2002)). How should an ASM deal with this issue ?

A first idea would be to use the time-stamp imprinted on each order. This information is particularly important if one wants to work on the time-distribution of orders. Unfortunately this idea is technically irrelevant. The time-stamp of standard operating systems is given using the third decimal places of seconds. However, it is perfectly possible to process several orders in one out of one thousand second. The time stamp is therefore necessary, but can be under no circumstances used as an ID.

Agents, human-beings or artificial traders, do not have to fix orders' ID. This task is up to the order-book itself. The order-book must stamp orders during acceptance, mainly to avoid fraudulent manipulations from agents. One platform-ID refers to one order. This latter must be different from any other possible identification number indicated in the order file or corresponding to a time-stamp. This means that the platform can handle three different identifiers, which makes its structure rather complex. Nevertheless, this additional complexity is mandatory if one wants to be able to use the ASM as a replay-engine and with artificial agents as well.

3.3 The scheduling system

The scheduler is a particularly critical element in all multi-agents systems. This component manages the very moment and situations in which agents have word (act), hence provides us complete control of agents actions. The scheduling system aims at avoiding possible biases in the simulation. However, this fundamental component in MAS systems is seldom discussed.

Outside the Computer Science community, it is often believed that using independent processes for each agent is a guarantee of autonomy. This is definitely not the case. Using threads consists in letting the operating system scheduler decide which agent will have the word at the next step in the simulation. Another delusion is to believe that threads will allow agents to work in parallel: using a single processor, there is necessarily one and only one single process running at each time step. Parallelism is simply simulated by the operating system. Nevertheless,the main disadvantage of this approach is that the system scheduler does neither exclusively consider agents nor even the MAS application itself; it also manages all applications running on the computer. Therefore, except on specialized real-time systems, there is no chance to observe agents solicited exactly at the same (relative) moment when two executions of the same simulation are processed. Results cannot be reproduced perfectly and the developer loses command on the code execution. It is therefore mandatory to code a specific scheduler to avoid these shortcomings.

3.3.1 When can the agents express their intentions?

One should not desire performing a loop in the simulation that keeps the word order among agents unchanged. This would introduce biases in the simulation : the first chosen agent would have systematically a priority over other agents; the last one might wait a long time before being allowed to express its intentions. Performing a uniform randomization of agent's word would lead to the same issues as well. In this last case, a few agents can theoretically stay unselected for a long time and even be ignored by the system.

Simulations in ATOM are organized as "round table discussions" and are grounded on an *equitably random* scheduler. Within every "round table discussion", agents are randomly interrogated using a uniformly distributed order. This latter feature ensures that each of them has an equal *possibility* of expressing its intentions. Notice that the API offers a random generator that is shared by all agents. The reproducibility of experiments is therefore guaranteed : one can either use a seed during the initialization of an experiment, or use ATOM in the "replay-engine" configuration since, as mentioned before, any simulation delivers a record of all the orders.

3.3.2 How do the agents proceed?

In real life, investors do not share the same attitudes. Different agents may have different time scales - some can make decisions every day and others every month of every second. Time scale of traders is a critical component in agent-based simulations. It affects such aspect of the simulation as repeatability (replaying) of the experiments. In ATOM this problem is solved in such way: a possibility to express an intention does not necessarily imply that a new order is issued. Since agents are autonomous, they always have the possibility to decline this opportunity. Developing an agent that sends twice less orders than any other agent can be made in programming her behaviour such as she will decline word on odd turns (keep unchanged position), while others accept it each time they have the possibility to do so. Let consider that one trading day contains 2000 ticks (rounds), thus an agent that trade once per day due to her strategy will accept a possibility to act only once over 2000. An agent trading once per month will accept a possibility to send an order once over 2000×22 (22 business days).

Moreover, if an agent would been allowed to send several orders when interrogated, this would lead to an equity problem similar to the one described before. To overcome this issue, we introduce *"one order to each book"* rule: agents are just allowed to send at most *one single* order to *a given order book* (*i.e.* one order at most per stock) within the same "round table discussion". However, if an agent plans to issue several orders concerning the same stock (thus, the same order book), she must act as a finite state automata. Each time she is allowed to express herself, she will change her state and send a new order. Developers can use this technique to set up various experiments without sacrificing a fair equity between agents or a perfect reproducibility of their protocols.
However, notice that agents have the possibility to send several orders within the same "round table discussion" : this ability is simply constrained by the "one order to each book" rule. If the ASM is settled such as it runs a multi-stock experiment, an agent can therefore rebalance her portfolio using one order per category of stocks she holds. The proper system scheduler provides this possibility.

3.4 Human in the loop

Current situation, when software agents are commonly used to replace human agents in making decision and taking action in the electronic trading, requires software solution to place human and artificial agents together in order to obtain advanced interaction. ATOM can include human-beings in the simulation loop. This is an important feature that is seldom offered in multi-agent artificial stock markets, if simply possible with respect to the algorithmic choices made in other platforms. A human agent is an interface allowing for human-machine interaction. Through this interface one can create and send orders. Notice that human agents do not have any artificial intelligence: they just embed human intelligence in a formalism that is accepted by the system.
To allow the introduction of human in the loop, ATOM has been designed to deal with communications over the network. Human agents can be run on different machines and the system allows client-server configurations. This approach is particularly fruitful for a pedagogic use of the platform during Finance class for example. In this latter case, several students have their own trading interface on their computers. In other terms, each of them runs a human agent linked to the ATOM server through the network. However, the presence of human agents does not alter the way the scheduler operates.

Two kinds of human agents can co-exist in ATOM: Modal Human Agents (MHA) and Non Modal Human agent (NMH).

- MHA can stop the scheduling system. As long as human-entity does not express her intentions (to issue a new order or to stay unchanged), the simulation is temporary frozen. In a classroom, this aspect is particularly important and leaves time for students to estimate current position and to make decision.

- NMH cannot freeze the simulation, which means that human agents compete in real time with artificial traders. Even if human agents can have a hard time in this situation, it remains realistic in a financial world where algorithmic trading is more and more frequent.

In this section we have presented three major technical points that characterize ATOM and should also concern many ASM. Even if other important technical issues could not be mentioned in this article, we have stressed that the development of artificial stock market platforms put forward a series of complex issues in terms of computer science. In the next section, we introduce some additional elements relative to the artificial intelligence of virtual agents that can be run in our platform. This question is of main concern for computer scientists and for financial researchers alike.

4 Artificial traders: from basic reactive agents to highly sophisticated entities

Artificial agents, market participants, comply with basic agent-based modeling concepts (Wooldridge and Jennings, 1995).

- *Autonomy* means that an agent is not passive subject to a global, external flow of control in its actions. An agent has own objectives, abilities to accept information, then to analyze it and based on these results to make decision about further actions.
- *Proactivity* means that the agents act in order to achieve its objectives or goals. In terms of artificial financial market, agents trade (set up the buy and sell orders) to maximize their wealth.

Many ASM can run large populations of homogeneous, respectively heterogeneous artificial traders. This is also the case for ATOM, moreover it allows facilities which are not available in other platforms. Generally speaking, artificial traders are characterized by their a) available set of *actions* (buy, sell) and possibility to switch between these activities (from buyer to seller) b) *decision making rules*, for instance, buyers cannot buy at a price higher than their buyer value and sellers cannot sell for a price below their seller cost c) *scheduling* of action: how often agent is able to send the orders in respond to market request, some agent participate one time per hour, while others trade every minute d) *information* consideration, in the mean which information agent requires from market or external word in order to make decision and what kind of information she shares for others e) possibility to describe *status* in mean of number of assets and available cash or current budget. Agents heterogeneity is driven by different combinations of these properties. For example, the following types of agents can be implemented:

Zero Intelligence Traders (ZIT): This behavior is merely based on stochastic choices: there are equal possibilities to send ask or bid order, ZIT do not observe and do not ask any information to set up prices and quantities, that are random variable. Concerning scheduling, such traders respond to every market request. This kind of behavior has been popularized in economics by (Gode and Sunder, 1993). Despite their extreme simplicity, these agents are widely used because more sophisticated forms of rationality appear to be useless to explain the emergence of the main financial stylized facts at the intraday level.

Technical Traders: "Chartists" are a specific population of technical traders. These agents try to identify patterns in past prices (using charts or statistical signals) that could be used to predict future prices and henceforth send appropriate orders. One can find an example of such behavior in (Arthur, 1994). From a software engineering perspective, these agents need to have some feedback from the market and some kind of learning process as well (reinforcement learning for large sets of rules is generally used). At the same time, technical traders ignore the actual nature of the company, currency or commodity. This lead to some complex algorithmic issues. For example, if one considers a population of a few thousand Technical Traders, it is highly desirable to avoid that each agent compute the same indicators, or simply store themselves the whole price series.

Sophisticated Intelligence Traders (SIT) : Several kinds of SIT can evolve in ATOM:

i) *Cognitive Agents* generally have a full artificial intelligence, although it can be designed to be rather minimal (usual features to develop such agents are memory, information analysis processes, expectations, strategies and learning capacities). For example, an agent buying at a specific price and sending immediately a "stop order" to short her position if the price drops under θ% times the current price, will fall in this category. Agents using strategic order splitting (see for example (Tkatch and Alam, 2009)) or exploiting sophisticated strategies (for instance, (Brandouy et al., 2009)) can also be considered as Cognitive Agents.

ii) *Evolutionary Agents* are the ultimate form of SIT; they outperform Cognitive Agents in terms of complexity since they are able to evolve with their environment. These agents can also generate new rules or strategies (this can require genetic algorithms for example).

iii) *Mean-Variance Agents* are investors trading over several order books, hence refer to portfolio optimisation aspects. When an investor wants to reoptimize her portfolio, she chooses and "ideal" portfolio from a mean-variance efficient frontier (Markowitz, 1952), that is based on analysis of internal and external information. The choice of portfolio depends on the trader's risk aversion. These agents send buy and sell orders in order to get closer to "ideal" portfolio. Such population of the agents is heterogeneous due to their initial cash available, reoptimization (trading) frequency and risk aversion.

Each and every of these agents can be implemented in ATOM. Notice again that if each kind of agents can be mixed with others, ATOM also allows for human beings to be added into any artificial stock market through a HMI.

5 Validation tests

As mentioned previously (see section 2), every ASM should succeed in processing perfectly a given order flow collected from a real-world stock market at a specific date. The result is obtained confronting prices delivered by the market at this date and the prices generated by the ASM using the same set of orders. It should also generate relevant "stylized facts" with regard to their real-world counterpart : these stylized facts are statistical characteristics of financial time series that prove to be systematically observed in various contexts (different assets, periods of time, countries).

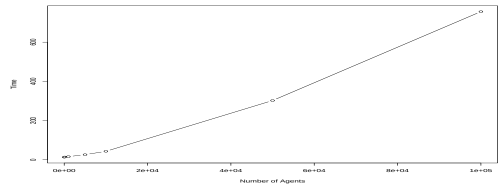
This section presents how ATOM fulfill this requirements, moreover, performance tests are considered.

5.1 Performance test

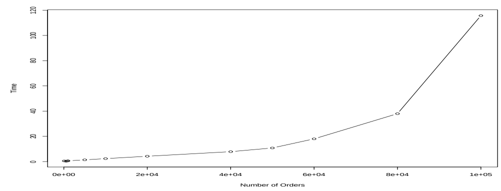
We run several experiments to demonstrate running time for realistic price series generation and existing order-flow execution.

To demonstrate ATOM price fixing ability, we use a group of heterogeneous agents. The population consists of Zero Intelligence Traders (ZIT) and Technical Simple Moving Average Traders (in the equal proportions), described in the section 4. Number of fixing prices is 10^5 [1]. Number of agents varies from 10 to 10^5. Results are introduced in the Figure 3(a). It takes about 12 minutes to run 10^5 agents for price fixing.

To test running time of replaying engine, we use real market order flow. The same agents population is used to read all variety of orders (limit, market, stop-limit, iceberg, etc.) and send them to order book. It is up to the market to fix price in a proper way (according to a fixing protocol). Number of orders vary from 100 to 10^5. It takes 2 minutes to replay 10^5 orders (see figure 3(b)).



(a) Price fixing time



(b) Orders flow execution time

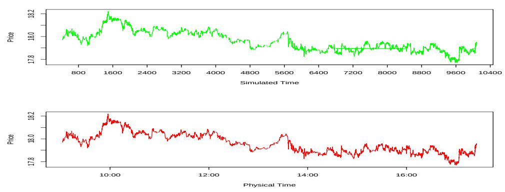
Figure 3: Results of the performance testing

5.2 ATOM reality-check

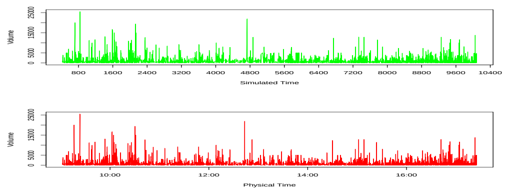
In this section, we report a series of tests conducted to check whether ATOM can generate financial dynamics in line with the ones of the Euronext-NYSE stock-exchange or not. The first series of test is devoted to the ability of ATOM at generating unbiased prices when it deals with a real-world order-flow.
Figure 4(a) and Figure 4(b) report results of the first reality-check (top Figures report results produced with the ATOM data, bottom Figures being those based on Euronext-NYSE data). We ran ATOM with a Hollow Agent reading the entire set of 83616 orders concerning the French blue-chip France-Telecom (FTE) recorded on June 26th 2008 between H9.02'.14".813"' and H17.24'.59".917"'. As mentioned previously, handling time in simulations is particularly complex and may lead to unsolvable dilemma. We cannot guarantee an exact matching of waiting times but rather a coherent distribution

[1] On the Euronext Stock Exchange the number of fixed prices for different stocks varies from 1000 to 5000 per day

of these values delivered by the simulator engine with regard to the observed waiting times.



(a) Prices : ATOM vs. Euronext-NYSE



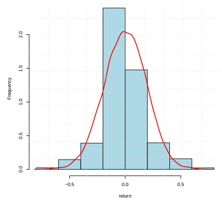
(b) Volumes : ATOM vs. Euronext-NYSE

Figure 4: Results of the "Reality Check" procedure

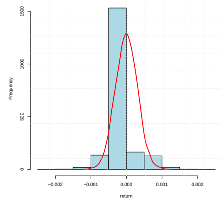
Notice that ATOM performs rather decently in satisfying the first reality check procedure.

5.3 Stylized facts

The second subset of tests focuses on ATOM ability to generate realistic artificial prices when populated with artificial agents. We ran a series of simulations to verify if ATOM can generate major stylized facts that are usually reported in the literature (see for example (Cont, 2001)). For the sake of simplicity and space-saving, we only report in a pictorial form of the classical departure from Normality of asset returns at the intraday level (Figures 5(a) and 5(b)). Notice again that these statistics are reported on the left-hand Figures when based on ATOM prices and on the right hand when based on Euronext-Nyse data. ATOM produces stylized facts, quantitative as well as qualitative, that is quite difficult task for most of artificial market platforms. Even if some artificial markets are able to reproduce the main stylized facts such as the non Gaussian return distribution or volatility clustering, the corresponding quantitative characteristics (basic statistics) do not fit real ones. ATOM can be easily calibrated to match specific quantitative market features (moments). This calibration facility is described in detail in the paper (Veryzhenko et al., 2010).



(a) Departure from Normality, ATOM



(b) Departure from Normality, Euronext-NYSE

Figure 5: Stylized facts, ATOM vs. Euronext-NYSE

6 Possible interactions within ATOM

There are two important points of interaction in the financial market:

- Who is interacting with whom: market-market, agent-market, agent-agent, agent-external world, etc.
- What are the results of interaction: monetary payoffs, the commodity, money. Let's consider, agent A interacts with agent B, provides information to agent C and pays money to agent B.

In this paper we consider only the first issue. Existing empirical studies of stock markets pay attention to the *market-market* relation. We observe the long-run linkages and co-movements between the markets, such as Canadian, Mexican and United States. At this moment, such interaction is not developed in ATOM, but can be realized due to possibility to incorporate the different market micristuctures in the *Markets* modular.

It is well-known facts that market's infrastructure and rules is an important impact to the strategies of participants and profitability of these strategies, that is *agent-market* interaction. Traders submit orders depending on the state of the order book or best quotes. These orders may result in a change in the best prices. The state of the market changes over time. The submission of an order is then dependent on the state of the market at the time of the submittal. A feedback loop is formed: a trader submits orders which affect the state of the market which affects the decisions of the trader on what order to submit. This kind of interactions is observed in the ATOM, through Euronext Stock Exchange sets of rules and agents behavior, in other worlds, interaction between *Markets* and *Agents* modulars. It exactly relies to *MIC**, where the environment defines actions sets of autonomous agents to achieve their goals. Moreover, existing models of advance traders interactions fit well *MIC** architecture. Agents interact with one another in order to achieve either a common or individual objectives. Let's consider some of existing "agent-agent" interaction approaches.

i) *The simplest agent-agent interactions* is "communication" through environment, in the term of stock market, through market microstructure organization. In such interactions, agent have different access to external information and different interpretation of it, they estimate own status and made decision about perspective bid-ask prices and don't share the trading strategy. In this way, agent i sends her bid and ask orders with their prices β_i and α_i respectively, with bid-ask spread $\alpha_i - \beta_i$. A trade can be concluded between agents i and j, if $\beta_i > \alpha_j$, and one chooses agents propose the maximal bid and minimal ask price. In order to keep agents equality and to avoid the biases in the internal information access, agents should be informed about book order changes simultaneously. Following notification method is realized in ATOM: all orders, as influences, are collected in the order book, once, all agents have sent their orders, price is fixed as reaction. This model fits IRM4S concept. As result, the strategies influence general system state (price formation) through "collection of influences" (order book), at the same time, agents use current environment information (historical price) for further decisions, in other worlds, the agents can also interact through the common variable of the past price history, but they are not directly affected by the actions of others.

ii) More *complex communication model* has been introduced by Cont and Bouchaud (Bouchaud and Potter, 2000) (a special case of Ising Model). Agents have three choices of market action: buy, sell, unchanged. They can form coalitions with other agents who share the choice of action. N agents are assumed to be located at the vertices of a random graph, and agent i is linked to agent j with a probability p_{ij}. A coalition is simply the ensemble of connected agents (a cluster) with a given action $\triangle\Theta$. Agents in a cluster share the same actions and do not trade among themselves.

iii) In the *MIC** model two agent considered as interaction when perceptions of an agent are influenced by the emissions of another. Let's consider *adaptive populations model* of Lux and Marchesi (Lux and Marchsi, 1999), where agents are divided into two groups: fundamentalist and noise(chartist) traders. Fundamentalist traders believe that stocks have a fundamental price p_f. Consequently, they sell stocks if the price $p(t)$ is higher that fundamental price and buy in the opposite case. Noise traders rely on chart analysis techniques and use behavior of other agents as information sources. Moreover, noise traders are divided into an optimistic and a pessimistic group. When the stock price rises, optimistic noise traders will buy additional shares while pessimistic noise traders will start selling. The important feature of this model is possibility to switch strategy between optimistic and pessimistic patterns, moreover between the noise and the fundamentalist agents groups, based on the profit difference in such groups. Generally speaking, agents can move between trading groups (noise, fundamentalists) after the computation of trading outputs in the mean of profit, as result, they change calculation or decision-making technique, this rule relies to *MIC** scheme. Cross-group movements change the traders group proportion, hence change the market state, that is influenced also by individuals' behavior (market price dynamic depends on agents strategies).

iv) The *direct interaction* is introduced by Kirman (Kirman, 1993), an influence opinion formation model. Agent's may hold one of two views. In each time step, two agents may randomly meet, and there is a fixed probability that one agent may convince the other one to follow his opinion. In addition, there is also a small probability that an agent changes his opinion independently. Applied to a financial market setting, one may therefore observe such interaction, like in the previous model, between technical traders and fundamentalist, that drive the market dynamics. Note that agents may change rules due to direct interactions with other agents, but switching probabilities are independent of the performance of the rules.

This is only some agent interaction models, that are realized or can be developed under ATOM API, this possibility is provided in the *Agents* components, where populations of agents and their relations are defined.

Agent-economic world is an important interaction, when agent seek the asset portfolios management. This is one of the flexibilities proposed in the ATOM API. Several assets can be traded at each time step by any-kind of agent, using sophisticated or extremely basic strategies (such as optimal, respectively naive diversification). In this case, agents should use information about the state of the artificial economy at a given time horizon. All these information are provided by the *Artificial Economic World* component, which adds the possibility of describing the complete set of temporal dimensions in the system (past, present, future). In this latter case, agents could eventually use past prices for the computation of comoments among assets and future values for expected returns and volatility, or they accept these "news" from *Artificial Economic World* as parameters, in order to employ mean-variance rebalancing theory.

7 Conclusion

The recent financial crisis has stressed the need for new research tools that can deal with the high level of complexity of the economic world. Agent based methods propose a powerful alternative to traditional approaches developed in finance. Among others, Artificial Stock Markets offer a completely controlled environment to test new regulations, new exchange structures or new investment strategies.

We showed that to build a realistic artificial stock market platform is a difficult task, but can be easily realized using main MAS concepts: agents behaviour, environment etc. We also discussed a series of software engineering and architecture design issues arising when the ultimate goal is to develop a complete API for market simulation. The purpose of the current work is to provide a polymorphic platform: it therefore can be used for a wide range of large scale experiments, including or not artificial agents, sophisticated behaviors, communication over the network... The possibility to employ well known MAS abstract models in the financial market modeling has been considered. We have presented how these notions have governed the development of the ATOM (ArTifical Open Market) API.

REFERENCES

Arthur, B. (1994). Inductive reasoning and bounded rationality : the el-farol problem. *American Economic Review*, 84:406–417.

Bouchaud, J.-P. and Potter, M. (2000). *Theory of Financial Risk*. Cambridge University Press.

Brandouy, O. and Mathieu, P. (2007). A conceptual framework for the evaluation of agent-base trading and technical analysis. *Artificial Markets Modeling. Methods and Applications. Lecture Notes in Economics and Mathematical Systems*, 599:63–79.

Brandouy, O., Mathieu, P., and Veryzhenko, I. (2009). Expost optimal strategy for the trading of a single financial asset. *SSRN eLibrary*.

Cesa-Bianchi, N. and Lugosi, G. (2001). Worst-case bounds for the logarithmic loss of predictors. *Machine Learning*, 43:247264.

Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1:223–236.

Ferber, J. and Muller, J.-P. (1996). Influences and reaction: a model of situated multiagent systems. *Second International Conference on Multiagent Systems ICMAS-96*, pages 72–79.

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

Gouaich, A., Michel, F., and Guiraud, Y. (2005). Mic : A deployment environment for autonomous agents. *E4MAS 2004, LNAI 3374, Springer-Verlag*, pages 109–126.

Inchiosa, M. E. and Parker, M. T. (2002). Overcoming design and development challenges in agent-based modeling using ascape. (3):7304–7308.

Jacobs, B. I., Levy, K. N., and Markowitz, H. M. (2004). Financial market simulation. *The Journal of Portfolio Management*, 30th Anniversary Issue:142–151.

Kirman, A. (1993). Ants, rationality, and recruitment. *Quarterly Journal of Economics*, 108:137–156.

Le Baron, B. (2002). Building the santa fe artificial stock market. *Working Paper, Brandeis University*.

Lux, T. and Marchsi, M. (1999). Scaling and criticality in a stochastich multi-agent model of a financial market. *Nature*, 397:498–500.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1):77–91.

Michel, F. (2007). The irm4s model: The influence/reaction principle for multi-agent based simulation. *AAMAS'07. Sixth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 908–910.

Mitchell, T. M., editor (1997). *Machine Learning*. WCB/McGraw-Hill, New York, NY.

Raberto, M., Cincotti, S., Focardi, S., and Marchesi, M. (2003). Traders' long-run wealth in an artificial financial market. *Computational Economics*, 22:255–272.

Ricordel, P.-M. and Demazeau, Y. (2001). Volcano, a vowels-oriented multi-agent platform. In *CEEMAS*, pages 253–262.

Tkatch, I. and Alam, Z. S. (2009). Strategic order splitting in automated markets. *SSRN eLibrary*.

Veryzhenko, I., Brandouy, O., and Mathieu, P. (2010). Agent's minimal intelligence calibration for realistic market dynamics. *Progress in Artificial Economics Computational and Agent-Based Models. Lecture Notes in Economics and Mathematical Systems 645*, pages 3–14.

Witkam, J. (2003). Altreva adaptive modeler.

Wooldridge, M. and Jennings, N. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152.

Woolridge, M. (2002). *Introduction to Multiagent Systems.* Introduction to Multiagent Systems., New York, NY, USA.

Zambonelli, F., Jennings, N. R., and Wooldridge., M. (2003). Developing multiagent systems: The gaia methodology. *In ACM Transactions on Software Engineering Methodology*, 12(3)(3):317–370.