A Generic Architecture for Realistic Simulations of Complex Financial Dynamics

Philippe MATHIEU[1] and Olivier BRANDOUY[2]

[1] Université des Sciences et Technologies, Computer Science Dept. & LIFL (UMR CNRS-USTL 8022), philippe.mathieu@lifl.fr
[2] Sorbonne Graduate School of Business, Dept. of Finance & GREGOR (EA MESR-U.Paris1 2474), olivier.brandouy@univ-paris1.fr

Artificial Stock Markets (here after ASM) have received an increasing amount of academic interest since the seminal works of [20] or [17]. Such platforms have benefitted from advances and new methods developed in the field of multi agent-systems (see for example [10], [14] and [27]). These Agent-Based virtual environments are particularly useful to study various aspects of the financial world in an entirely controlled environment, opening new perspectives for policy makers, regulatory institutions and firms developing business solutions in the financial industry (for example asset management or trading). There is little doubt ASM could become a key system in the post financial crisis risk-management toolbox to overcome the weaknesses of traditional approaches. Agents-based modeling and simulation offer frameworks to study the impact of a Tobin's tax for example, or to develop new stress tests for assessing financial resilience to economic shocks or to develop new automatic trading techniques.
In this research paper, we introduce a new, highly flexible Agent-Based model of financial markets in an API form. This application offers a solution for implementing realistic simulations of complex financial dynamics using both artificial intelligence, distributed agents and realistic market algorithms. We consider that if various questions in Finance can be solved with Agent-Based Modeling solutions, Multi-Agent Systems directly benefit from Financial questions as well : ASMs, with driving simulators (see [9]), offer one of the richest environments to evolve software engineering for multi-agent systems . Therefore, the punchline of this paper is that development of artificial markets offers the whole variety of issues one can face in agent based modeling. Among others, ASM are grounded on an individual-based approach with local interaction, distributed knowledge and resources, heterogeneous environments, agent autonomy, artificial intelligence, speech acts, discrete scheduling and simulation. We defend this punchline with an illustration derived from the building of the ArTificial Open Market API (here-after ATOM, see http://atom.univ-lille1.fr).

This paper is organized as follows. In a first section we briefly present the relevant literature around artificial markets and Agent-Based modeling in Finance. In a second section, we extensively introduce the ArTificial Open Market API and present the mains computing issues linked to its development. In the third section, we present agent's behavior and introduce a series of tests geared at verifying the efficiency of our API. We then conclude and open new research and technological perspectives.

1 Elements of literature in Agent-Based finance

There is a long and fruitful research tradition in Economics based on the so-called ”methodological individualism” which could be referred as to the ”individual-based approach” in computer science (see for example [24]). Methodological individualism proposes a reductionist perspective for the analysis of macroscopic economic phenomena. Along this approach, these latter must ultimately be explained by agents' preferences and actions. The complex system approach initiated by [1] in physics and [2] and [3] in economics has profoundly renovated this philosophy. This approach has also altered the common comprehension of financial markets built by neoclassical economists (see [5]). In this perspective, Agent-Based models of financial markets were developed to renew the analysis of various ill-understood issues in Finance. Acknowledging that financial systems are evenly characterized by order and disorder, that grasping economic complexity could not be done easily with traditional techniques, a few researchers used Agent-Based models and simulations to study financial markets emergent dynamics (for instance, [13] or [20]).

Developing ASM necessarily implies to select components in the model that are ex-ante considered as necessary from a theoretical point of view. On the one hand, a fine selection of these elements is definitely critical and should be conducted keeping in mind the Occam's razor principle. On the other hand, ignoring some essential parameters in the model could strongly weaken its relevance. For example, in Agent-Based Finance, market microstructure, which refers to the way transaction are organized, is often reduced to a mere equation linking price formation to the imbalance between supply and demand (see, among others [16], [7] or [11]). This simplification erases one important feature in real-world markets : trading is generally not synchronous.Other approaches have overcome this issue introducing directly in the Agent-Based model an order-book and a scheduler organizing non-synchronous trading. Generally speaking, these models do not accept the whole variety of orders and only focus on limit and market-orders (for instance [21]) or cannot be used as a replay engine for existing order-flows (for instance [19]). They also suffer from a lack of flexibility and must be viewed as software rather than as APIs : this means that they perform quite well in doing what they are supposed to do, 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 the following section we introduce the ArTificial Open Market API. Among others, we make a specific point on the ability of this new, generic package to overcome the issues mentioned previously.

2 The Artificial Trading Open Market API : general principles and distinctive features.

ATOM is a general environment for Agent-Based simulations of stock markets. It is based on an 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. This might be done using imbalance equations as evoked in section 1. In the case of ATOM, price formation is ruled by a negotiation system between sellers and buyers based on an asynchronous, double auction mechanism structured in an order book. Using this API, one can generate, play or replay order flows (whatever the origin of these order flows, real world or virtual agents population). It also allows distributed simulations with many computers interacting through a network as well as local-host, extremely fast simulations. ATOM can be used to design experiments mixing human beings and artificial traders. One of the main advantages of ATOM consists in its modularity. This means that it can be viewed as a system where three interacting main components interact: i) *Agents*, and their behaviors, ii) *Markets* defined in terms of microstructure and iii) the *Artificial Economic World* (including an information engine and, potentially, several economic institutions such as banks, brokers, dealers...). The

two first components can be used independently or together. Depending upon the researcher targets, the *Artificial Economic World* can be plugged or not in the simulations. For example, one can use the system for the evaluation of new regulation policies or market procedures, for assessing potential effects of taxes or new trading strategies in a sophisticated artificial financial environment. Thanks to its high modularity and its 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. This 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.
Six distinctive aspects of ATOM can be highlighted:

1. It can be used without any agent. One can directly send orders written in a text file (for example, a set of orders as it arrived on a given day, for a given real-world stock market) to each order-book implemented in the simulation. In this case, ATOM serves as a ”replay-engine” and simulations merely rely on market microstructure. It therefore runs really fast (an entire day of trading in less than 5 seconds).
2. Agents can be viewed as simple nutshells in certain cases : they only take actions they are urged to by a third party. 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. Beyond ”Hollow Agents”, ATOM can use various kind of sophisticated agents with their own behaviors and intelligence (see section 3). Thousands of these agents can evolve simultaneously, creating a truly heterogeneous population. Once designed, agents evolve by themselves, learning and adapting to their (financial) environment.
4. In any of the previous cases, ATOM generates two files; the first file records orders emitted by the agents, with the platform time stamp fixed at the very moment they arrive to a given order-book. Notice again this file can be used in a ”replay-engine” configuration. The second generated file collects prices resulting from the orders.
5. ATOM can mix human-beings and artificial traders in a single market using its network capabilities. This allows for 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 2.4).
6. Any artificial stock market should be tested rigorously to verify if it matches the following criteria: i) ASM must have the ability to replay perfectly an order flow actually sent to a given market with the same microstructure. The resulting price series (on the one hand, the ”real-world” one and on the other hand, the ”artificial” one) should overlap perfectly. ii) Given a population of agents, the ASM should generate stylized facts qualitatively similar to the market it is geared at mimicking. As it will be shown later, ATOM succeeds in both cases.

2.1 Artificial Stock Markets as complex, adaptive systems

Artificial Stock Markets are environments allowing 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 [23], 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.

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 these structures that appear to be representative of the 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).

2.2 A unique identity for Orders

In its simplest form, an order is a triplet constituted by a 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 may need to be able to process 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. To be able to replay this order-flow, a first difficulty consists in interpreting exactly the order flow as it is expressed in the real-world. If *"Limit Orders"* had been present in this record exclusive of any other order type, the sequentiality of orders would have been sufficient to guarantee a perfect reproducibility. Unfortunately, issued orders can be modified or deleted are in many markets. This implies one must be able to identify clearly which previously issued order an "Update" or a "Delete" order points to. Thus, a generic platform has to use a unique ID for orders. This is particularly important in situations where 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 [22]) or in the Santa-Fe ASM for example (see [15]).
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 devoted to the order-book itself. The order-book must stamp orders at reception, mainly to avoid fraudulent manipulations from agents. One platform-ID is therefore affected to each 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.

2.3 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. 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 misunderstanding consists in believing 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 in this approach is that the system scheduler does neither exclusively consider agents nor even the MAS application itself; it also manages all applications running in 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.

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.

How do they proceed ?

In real life, investors do not share the same attitudes. Some will be more reactive than others, or will implement more complex strategies leading to a higher rate of activity. In ATOM having the 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 behavior such as she will decline word on odd turns, while others accept it each time they have the possibility to do so.

Moreover, if an agent had been allowed to send several orders when interrogated, this would have led to an equity problem similar to the one described before. To overcome this issue, 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, provided the scheduler has offered the possibility to do so.

2.4 Human in the loop

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. Following the example of driving or crowd simulators (see for example [25] or [9]), human agents and virtual agents can evolve together. Human agents do not differ from artificial agents in their philosophy : they share the same general characteristics as other agents. The so-called ”vowels” approach is respected, even if `U` (for ”Users”) is subsumed by `A` – Agents –. 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 while running. As long as her human owner 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 to students for deciding their actions.
- 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.

3 Agents Behaviors and 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; it also develops one key element in the system that has not been introduced for the moment : agent's behaviors.

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

Many ASM can run large populations of homogeneous, respectively heterogeneous artificial traders. This is also the case for ATOM although the API allows for facilities which are not available in other platforms. Generally speaking, artificial traders are characterized by their initial endowments (financial resources and information), and by their artificial intelligence or behavior. For example, the following types of agents can be implemented :

Zero Intelligence Traders (ZIT): This behavior is merely based on stochastic choices (orders are composed of prices, quantities and directions treated as random variables, *i.e.* uniformly drawn in a bounded interval). This kind of behavior has been popularized in economics by [12]. ZIT constantly provide orders but neither extract information from the market nor from any other component of the ASM. 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 (see above, section 3.3). Notice that several types of ZIT can be coded, which question the "zero-intelligence" concept itself.

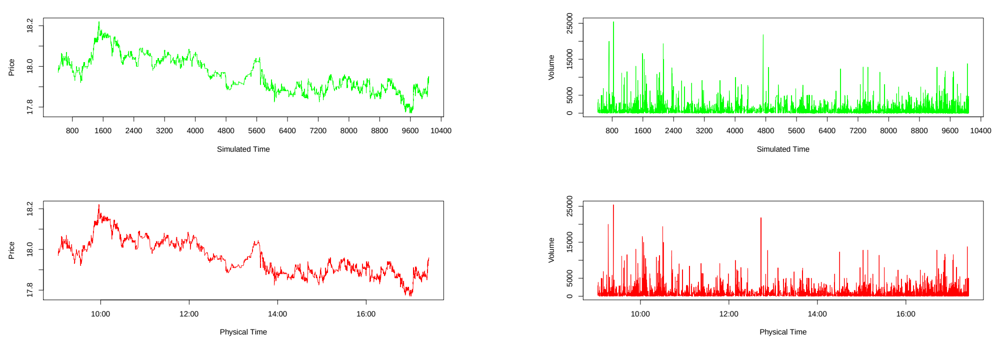
Technical Traders: "Chartists" are a specific population of technical traders. These agents try to identify patterns in past prices (using charts or statistical signals) that could be used to predict future prices and henceforth send appropriate orders. One can find an example of such behavior in [4]. 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). 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) : Three kind of SIT can evolve in ATOM:
i) *Finite-State Agents* and *Hollow Agents*; by construction one agent can only send one order per time tick ("round of word"). If this agent wants to send a series of orders to the same order book, this must be divided into several orders, one per time-tick. Finite-State Agents can deal with this issue, which implies a minimum level of sophistication in their behaviors. Hollow agents serve as a media to send orders formulated by other agents (human beings or artificial traders). These latter agents are especially useful for the replay-engine instantiation of ATOM.
ii) *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 $\theta\%$ times the current price, will fall in this category. Agents using strategic order splitting (see for example [26]) or exploiting sophisticated strategies (for instance, [6]) can also be considered as Cognitive Agents.
iii) *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). Each and every of these agents can be implemented in ATOM. Notice they all can manage asset portfolios if required. This is one of the flexibilities proposed in the 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 nave diversification). In these cases, 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 co-moments among assets and future values for expected returns and volatility (see [18]). 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.

3.2 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 1(a) and Figure 1(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 of these values delivered by the simulator engine with regard to the observed waiting times.



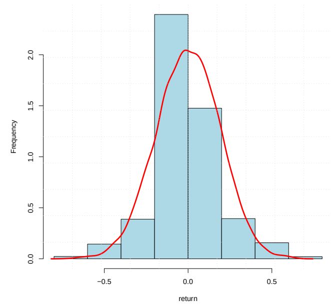
(a) Prices : ATOM vs. Euronext-NYSE (b) Volumes : ATOM vs. Euronext-NYSE

Fig. 1. Results of the "Reality Check" procedure

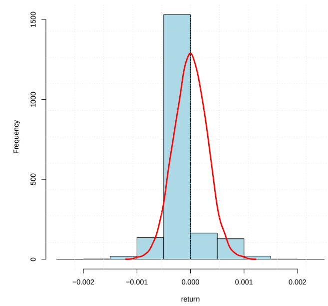
Notice that ATOM performs rather decently in satisfying the first reality check procedure.

3.3 Stylized facts

The second subset of tests focuses on the ability of ATOM at generating 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 [8]). 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 2(a) and 2(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. Real data are those used previously for the reality check, artificial data were generated using a population of ZIT as described in the ATOM API.



(a) Departure from Normality, ATOM



(b) Departure from Normality, Euronext-NYSE

Fig. 2. Stylized facts, ATOM vs. Euronext-NYSE

In clearly appears that ATOM produces stylized facts in line with those observed for a specific stock listed on Euronext-NYSE : this means that the platform can both generate realistic price series with a population of ZIT and process without bias a real order-flow.

4 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 such as econometric or statistical models. Among others, Artificial Stock Markets are particularly interesting in that they offer a completely controlled environment to test new regulations, new exchange structures or new investment strategies.
In this paper we have defended that these models offer one of the richest environments to illustrate multi-agent systems notions. We have particularly highlighted the importance of a polymorphic platform : it therefore can be used for a wide range of experiments, including or not artificial agents, sophisticated behaviors, communication over the network... We also discussed a series of software engineering problems arising when the ultimate goal is to develop a complete API for market simulation. A precise proof mechanism that can be used to validate any artificial market platform has been introduced. This proof is based on two tests : on the one hand, each platform should be able to replay perfectly a real set of orders and deliver exactly the corresponding real price series; on the other hand, while running with an artificial agents population, it should also generate prices that match the so-called financial ”stylized facts”.
We have presented how these notions have governed the development of the ATOM (ArTifical Open Market) API. This platform can evolve complex AI agents, the only limitation in this domain being researchers' imagination.

References

1. P. Anderson. More is different. *Science*, 177:393–396, 1972.
2. P. Anderson, K. J. Arrow, and D. Pines, editors. *The Economy as an Evolving Complex System*, volume 5 of *Santa Fe Institute Studies in the Sciences of Complexity*. Addison Wesley, Redwood City California, 1988.
3. B. Arthur. *Complex Systems*, chapter The Economy as a Complex System. Wiley, 1989.
4. B. Arthur. Inductive reasoning and bounded rationality : the el-farol problem. *American Economic Review*, 84:406–417, 1994.
5. B. Arthur. Complexity in economic and financial markets. *Complexity*, 1(1):20–25, 1995.
6. O. Brandouy, P. Mathieu, and I. Veryzhenko. Ex-post optimal strategy for the trading of a single financial asset. *SSRN eLibrary*, 2009.
7. S. Cincotti, L. Ponta, and S. Pastore. Information-based multi-assets artificial stock market with heterogeneous agents. In *Workshop on the Economics of Heterogeneous Interacting Agents 2006 WEHIA06*, 2006.
8. R. Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1:223–236, 2001.
9. K. M. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *J. Artif. Intell. Res. (JAIR)*, 31:591–656, 2008.
10. J. Ferber. Multi-agent system: An introduction to distributed artificial intelligence. *J. Artificial Societies and Social Simulation*, 4(2), 2001.
11. F. Ghoulmie, R. Cont, and J.P. Nadal. Heterogeneity and feedback in an agent-based market model. *Journal of Physics: Condensed Matter*, 17:1259–1268, 2005.
12. D. K. Gode and S. Sunder. Allocative efficiency of market with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, February 1993.
13. G. Kim and H. Markowitz. Investment rules, margin, and market volatility. *Journal of Portfolio Management*, 16(1):45–52, 1989.
14. Y. Kubera, P. Mathieu, and S. Picault. Interaction-oriented agent simulations: From theory to implementation. pages 383–387, 2008.
15. B. Le Baron. Building the santa fe artificial stock market. *Working Paper, Brandeis University*, June 2002.
16. B. Le Baron, W.B. Arthur, and R. Palmer. Time series properties of an artificial stock markets. *Journal of Economic Dynamics and Control*, 23:1487–1516, 1999.
17. H. Levy, M. Levy, and S. Solomon. A microscopic model of the stock market: Cycles, booms, and crashes. *Economic Letters*, 45(1):103–111, May 1994.
18. H. Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91., 1952.
19. L. Muchnik and S. Solomon. Markov nets and the natlab platform; application to continuous double auction. *New Economic Windows*, 2006.
20. R.G. Palmer, W.B. Arthur, J.H. Holland, B. LeBaron, and P. Tayler. Artificial economic life: A simple model of a stockmarket. *Physica D*, 75:264–274, 1994.
21. M. Raberto, S. Cincotti, C. Dose, and S. M. Focardi. Price formation in an artificial market: limit order book versus matching of supply and demand. *Nonlinear Dynamics and Heterogenous Interacting Agents*, 2005.
22. M. Raberto, S. Cincotti, S. Focardi, and M. Marchesi. Traders' long-run wealth in an artificial financial market. *Computational Economics*, 22:255–272, 2003.
23. P.-M. Ricordel and Y. Demazeau. Volcano, a vowels-oriented multi-agent platform. In *CEEMAS*, pages 253–262, 2001.
24. J. Schumpeter. On the concept of social value. *Quarterly Journal of Economics*, 23(2):213–232, 1909. http://socserv2.socsci.mcmaster.ca/ econ/ugcm/3ll3/schumpeter/socialval.html.
25. W. Shao and D. Terzopoulos. Autonomous pedestrians. *Graphical Models*, 69(5-6):246–274, 2007.
26. I. Tkatch and Z. S. Alam. Strategic order splitting in automated markets. *SSRN eLibrary*, 2009.
27. M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & sons, 2009. ISBN 978-0521899437.