# SIMULATING AN EVOLUTIONARY MULTI-AGENT BASED MODEL OF THE STOCK MARKET

**Diana MARICA**
*The Romanian Academy, 010071, Romania*
*diana.v.marica@gmail.com*

**Abstract**

*The paper focuses on artificial stock market simulations using a multi-agent model incorporating 2,000 heterogeneous agents interacting on the artificial market. The agents interaction is due to trading activity on the market through a call auction trading mechanism. The multi-agent model uses evolutionary techniques such as genetic programming in order to generate an adaptive and evolving population of agents. Each artificial agent is endowed with wealth and a genetic programming induced trading strategy. The trading strategy evolves and adapts to the new market conditions through a process called breeding, which implies that at each simulation step, new agents with better trading strategies are generated by the model, from recombining the best performing trading strategies and replacing the agents which have the worst performing trading strategies. The simulation model was build with the help of the simulation software Altreva Adaptive Modeler which offers a suitable platform for financial market simulations of evolutionary agent based models, the S&P500 composite index being used as a benchmark for the simulation results.*

**Key words:** *multi-agent based modeling; artificial stock market; genetic programming; heterogeneous agents; simulations.*

**JEL Classification:** *C63, G17*

## I. INTRODUCTION

A multi-agent model is a computational model for simulating the actions and interactions among artificial agents in order to analyze the emerging behavior and effects on a complex system as a whole, being a powerful tool in the understanding of markets and trading behavior. An agent-based model of a stock market consists of a population of agents which represent investors, and a price discovery and clearing mechanism representing an artificial stock market.

The aim of this research is to describe a multi-agent model of the stock market and to show the simulations results of the model incorporating 2,000 heterogeneous agents which trade within an artificial stock market. In order to achieve my research aim, the Adaptive Modeler software (Witkam, 2003) was used to simulate the multi-agent model as an evolutionary model of the stock market. Thus, heterogeneous agents trade a stock floated on the stock exchange market, placing orders depending on their budget constraints and trading rules, where the artificial market is simulated as a call auction market.

The evolutionary agent-based model simulates a single financial asset traded on the stock exchange, the maximum simulation steps in an epoch being 20,000. Each simulation step represents a trading day, or a trading session, which can be structured as a recurring cycle which is described in the next section. The agents are autonomous and heterogeneous entities representing the traders of the stock market, each having their own *wealth* and their own *trading strategy*.

The multi-agent based model is simulated under different computational techniques compared to other scientific papers in this field. This is due to the fact that in Strongly Typed Genetic Programming (STGP) (Montana, 2002) the process of estimating the agents' fitness function does not imply re-execution of the trading rules based on historical data, therefore there is no over-fitting when it comes to genetic programming induced trading strategies. This is possible due to the fact that the model evolves in a time-incremental way, and it does not optimize on historical data, thus avoiding over-fitting of the data which seems to represent one of the biggest forecasting pitfalls in this research field. The simulated model incorporates a high number of artificial agents, namely 2,000 agents which are endowed with 2,000 continuously adapting trading rules, which increases model stability and reduces sensitivity to random factors. The multi-agent model is dynamic, constantly evolving and adapting to market conditions.

To explore and understand the complexity of the financial markets and trading behavior, models using agent-based modeling techniques have been successfully implemented, offering explanation for observed stylized facts and being able to reproduce many of them (Aloud, Fasli et. al., 2013; Arthur, Holland et. al., 1997, LeBaron, 2006; Lux and Schornstein, 2005; Marchesi, Cincotti et. al., 2000). Arthur, Holland, et al. (1997) from

Santa Fe Institute, Ca., USA, developed an artificial stock market which allowed for testing of agent-based models with heterogeneous agents. Another important artificial stock market was developed by Marchesi, Cincotti, et. al. (2000), Raberto and Cincotti (2005) which is called Genoa Artificial Stock Market and uses a double auction trading mechanism similar with the one used in the hereto paper, the artificial agents trading in a random manner unlike the artificial agents from the model in this paper which use technical trading to assess the historical prices and to generate trading orders. The hereto model aims to forecast future prices by estimating the future trend from a simulated step to the next, by evaluating multiple trading strategies and following the aggregated behavior of traders. Other papers using adaptive modeling techniques for forecasting purposes are Leigh, Purvis et. al. (2002) which focus on forecasting the New York Stock Exchange composite index with genetic algorithms and Demir and Shadmanov (2015) which use neural networks to forecast economic growth.

Phelps, Marcinkiewicz et al. (2005), Holland and Miller (1991), Hommes (2006) have used in their experiments heterogeneous agents which change their strategies during the learning process, as follows: the unprofitable strategies are being replaced with the more profitable ones, thus developing adaptive models which use genetic algorithms to evolve. Allen and Karjalainen (1999), Walia and Byde (2003) have studied the development of the agent-based models which use genetic programming, allowing for more flexibility and effectiveness in finding optimal solutions, programs being encoded as tree structures, thus crossover and mutation operators being applied easier. The later is similar with the learning process used in the hereto paper.

This paper is structured as follows. Section II describes the multi-agent based model structure, Section III shows the simulation results of the model and Section IV ends the paper with the conclusions and directions for future work.

## II.MULTI-AGENT BASED MODEL STRUCTURE

The multi-agent based model described in this paper has two major components. The first one is represented by the artificial stock market in which agents trade, and the second one refers to the evolutionary process in the agent population which allows it to adapt constantly to the market conditions through genetic programming techniques.

The multi-agent based model has a 2,000 population of artificial and heterogeneous agents which during the initialization process are endowed with an initial wealth of 100,000 monetary units to use for investments. They are also endowed with a set of trading rules, the allocation of trading rules is done in a random manner so the agents have different trading rules, not all good or successful. At each simulation step, all agents evaluate the historical data series of a stock traded on the real market, they decide what percent of their wealth to invest or to sell and send the orders to the artificial market where the trading session is running. After the trading session is closed, all the agents performance is computed on a medium term and the worst performing agents are replaced with new agents which are generated by a combination of the best performing trading strategies. The later process is called breeding and is used with the scope of constantly improving the trading strategies of the agents in the population. This simulation step is equivalent with a trading session in the real market, and 20,000 of these simulation steps form a simulation epoch.

### 1. Artificial stock market

The artificial stock market trading mechanism is based on a call auction which represents an order driven facility which batches multiple limit orders together for simultaneous execution in a multilateral trade, at a single clearing price, at a predetermined point in time. A limit order is a price-quantity pair which expresses an offer to buy or sell a specific quantity at a specific price, while a market order specifies a quantity but not a price, limit order price being the maximum allowable bid or minimum allowable ask which allows the order to be executed. A single trader may submit a single order per batch interval, which are not visible to other agents during the batch interval, the auction being a sealed bid.

The trading mechanism used by the Adaptive Modeler software application to simulate the artificial stock market is set as call auctions mainly because many stock markets use this mechanism. In the call auction markets, agents introduce bid or ask orders, each order consisting of a price and quantity. The bids and asks orders received are put in the order book and an attempt is made to match them. The price of the trades arranged must lie in the bid-ask spread (interval between the selling price and buying price), which is a parameter of the simulated model. Also, all executed trades bear a trading cost as a variable broker fee of 0.20% of the transaction value. An example of the order book for the simulated trading activities is illustrated below in Fig. 1 for orders before the clearing process and in Fig. 2 for unexecuted orders remained in the market after the clearing process is finished. All the unexecuted orders are removed from the order book after every simulation step.

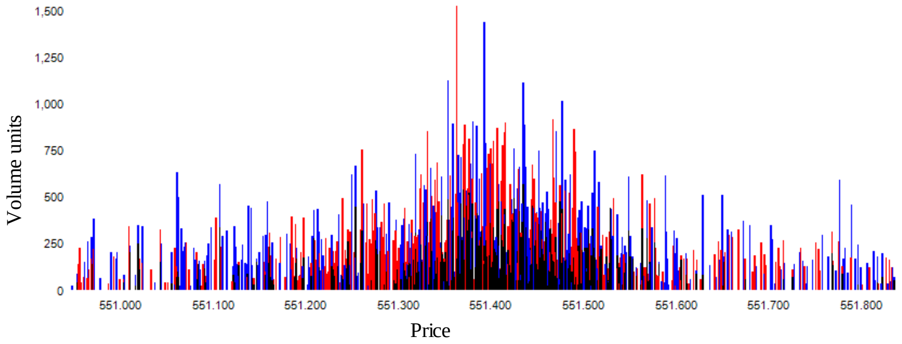

**Figure 1. Order book example from the multi-agent based model simulation. Blue bars represent bid volume orders, red bars represent ask volume orders, black bars represent bid and ask volume orders at equal prices, before market clearing.**

*Source*: Own simulation of the multi-agent based model done with Adaptive Modeler software application

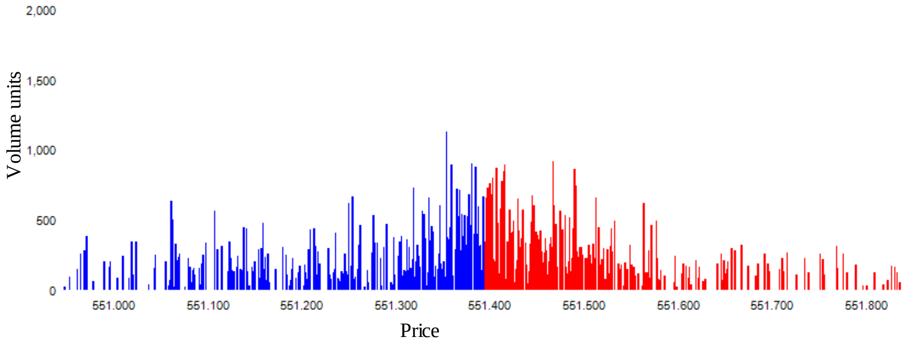

**Figure 2. Order book example from the agent-based model simulation. Blue bars represent bid volume from unexecuted orders, red bars represent ask volume from unexecuted orders, after market clearing.**

*Source*: Own simulation of the multi-agent based model done with Adaptive Modeler software application

## 2. Evolutionary process of the model

The genetic programming evolutionary cycle of each simulation step is summarized below:

1) *Receive new quote* bar from the real historical data quotation series of the S&P500 composite index.

2) *Agents evaluate trading rules and place orders*: Agents receive access to historical prices and evaluate the evolution of prices according to the technical analysis generated by their trading rules, resulting in a desired position as a percentage of wealth limited by the budget constraints, and a limit price. Agents are two-way traders during the simulations, meaning that they are allowed to both sell and buy during multiple simulation steps, and they are one-way traders during a single simulation step (in this case a trading session) corresponding to an auction, as they are able to submit only one order per auction, either buy or sell.

3) *Artificial stock market clearing and forecast generation*: The artificial stock market determines the clearing price in the call auction, which is a discrete time double-sided auction mechanism in which the artificial stock market collects all bids (buying orders) and asks (selling orders) submitted by the agents and then clears the market at a price where the supply quantity equals the demanded quantity. The clearing price is the price for which the highest trading volume from limit orders can be matched, thus all agents establish their final positions and cash at the same time. The artificial stock market executes all executable orders and forecasts the price for the next simulation period. The forecasted price is equal to the clearing price.

4) *Breeding*: During the breeding process, new agents are created from best performing agents in order to replace the worst performing agents, creating new trading strategies by recombining the parent trading strategies through a crossover operation, and creating unique trading strategies by mutating a part of the parent strategy. The breeding process repeats at each simulation step, with the condition that the agents must have a minimum breeding age of 80 simulation steps, in order to be able to assess the agents' performance.

In order to obtain random seed, the Adaptive Modeler software uses the Mersenne Twister algorithm (Matsumoto and Nishimura, 1998) to generate pseudo random number sequences for the initial creation of trading rules or genomes and for the crossover and mutation operators of the breeding process.

The trading strategy attached to each agent is called a genome and uses a decision tree composed of genes which are mathematical functions. The initial node in the genetic program tree combines the position desired in the security generated randomly, and the limit price value generated by a collection of functions working as a technical analysis on the historical prices, into a buy or a sell order advice. The desired position value ranges between -100% (short position, or selling position) and 100% (long position, or buying position). The limit price value is generated by a collection of functions which uses simple mathematical functions and technical analysis indicators initially generated in a random manner from the list of functions selected to be used in the model, which develop during the breeding process, in order to generate the limit price for the buy or sell order.

The agents' trading rules development is implemented in the software by using a special adaptive form of the Strongly Typed Genetic Programming (STGP) approach, and use the input data and functions that have the most predictive value in order for the agents with poor performance to be replaced by new agents whose trading rules are created by recombining and mutating the trading rules of the agents with good performance. In order to do this, a dynamic fitness function is used to evaluate the performance of the agents, being a metric of the agent's investment return over a certain period, therefore the Fitness Return function is computed as the wealth return over the last 80 analyzed quotes and represents the selection criterion for breeding.

Genetic programming was first developed by Koza (1992), while the special type of genetic programming used in the hereto paper is the Strongly Typed Genetic Programming technique which was introduced by Montana (2002), with the scope of improving the genetic programming technique by introducing data types constraints for all the procedures, functions and variables, thus decreasing the search time and improving the generalization performance of the solution found. The parameters of the model are described below in Table 1.

**Table 1. General settings of the models. Market and agents' parameters configuration**

| Type | Parameter Name | Parameter Value |
|---|---|---|
| Market Parameters | No. of simulation steps | 16,000 |
| | Imported data series | S&P500 index, Jan. 3rd, 1950 - Jan. 2nd, 2015 |
| | No. of agents | 2,000 |
| | Minimum price increment | 0.01 |
| | Average bid/ask spread | 0.01% |
| | Variable Broker fee | 0.20% |
| Agent Parameters | Wealth Distribution | 100,000 initial wealth for each agent |
| | Min. position unit | 5% |
| | Max. genome size | 1,000 |
| | Max. genome depth | 20 |
| | Min. initial genome depth | 2 |
| | Max. initial genome depth | 5 |
| | Genes | CurPos, RndPos, LevUnit, Rmarket, Cash, Bar, PndPos, IsMon, IsTue, IsWed, IsThu, IsFri, close, bid, ask, average, min, max, >, change, +, dir, isupbar, upbars, pos, lim, Advice, and, or, not, if |
| | Breeding Cycle Length | 1 simulation period |
| | Minimum breeding age | 80 simulation periods |
| | Initial selection: randomly select | 100% of agents of minimum breeding age or older |
| | Parent selection | 5% agents of initial selection will breed |
| | Mutation probability | 10% of the generated offspring will mutate |

## III. SIMULATIONS

The parameters described in Table 1 are used for the simulations of the S&P500 composite index for a period of 65 years, during Jan. 3rd, 1950 - Jan. 2nd, 2015, meaning 16,256 simulation steps were processed during the simulation epoch. The model forecasts the price from a simulated step to the next, and the agent trade based on the historical data series of S&P500 previous to the current simulation step, therefore they don't have the entire data at their disposal. As shown in Fig. 3, the simulation generates a forecasted price series with low errors from the benchmark which is represented by the real data series. The first 35 years of the data are used for training of the agents, therefore only results for the last 15 years are illustrated, mainly for visualization reasons.

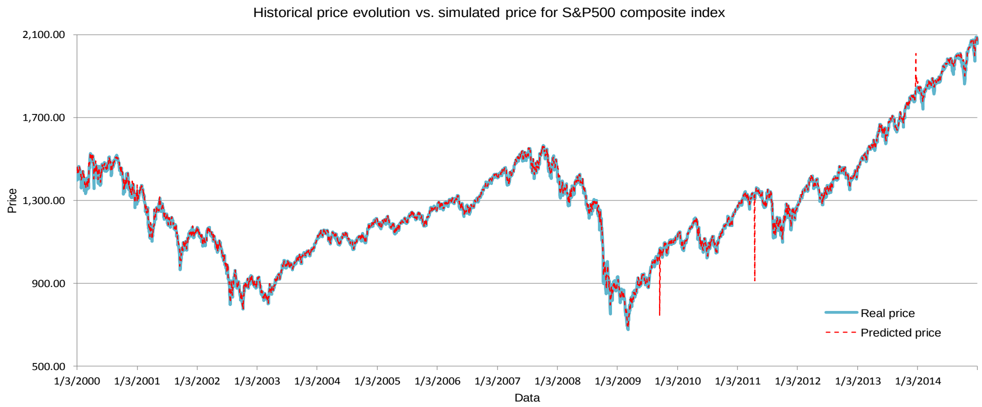

**Figure 3. S&P500 composite index price from real market vs. forecasted price**

*Source*: Own simulation of the multi-agent based model, forecast results obtained with Adaptive Modeler software application

Fig. 3 also shows a few spikes in the simulated data, which were generated by extremely low ask/bid trading volume, which can also appear in the real market and be interpreted as unresolved technical system failure coming from over-flooding the markets with buy/sell orders. Despite the few spikes in the simulation, the model shows stability which brings the simulated price back to being close to the benchmark.

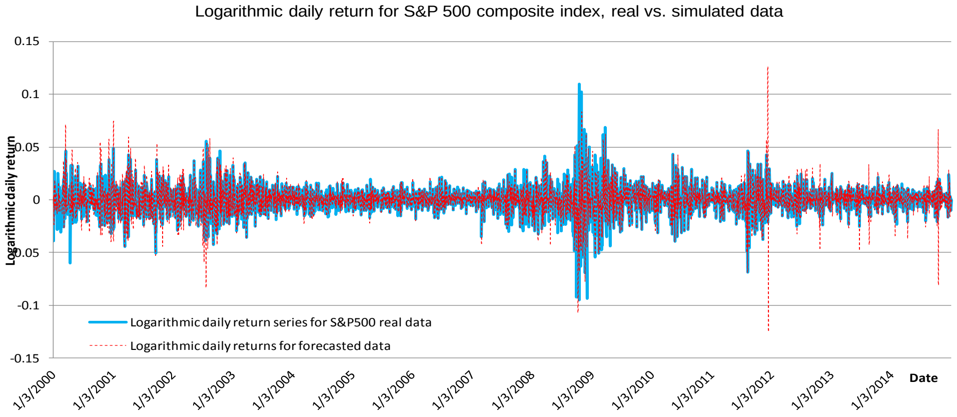

**Figure 4. S&P500 composite index logarithmic daily returns on real market vs. forecasted data**

*Source*: Own simulation of the multi-agent based model, forecast results obtained with Adaptive Modeler software application

In Fig. 4, the simulated logarithmic daily return is depicted side by side with the logarithmic daily return of the benchmark, the similarities of the dynamics and behavior of the two data series being staggering. Spikes in the simulated data reached a maximum of 13%, which is acceptable given that a 15% daily increase/decrease in stocks represents the threshold for many stock markets.

Fig. 5 shows consistent results as previous data, the volatility of daily returns of the data series being computed for both simulated and real data, as absolute daily logarithmic return.

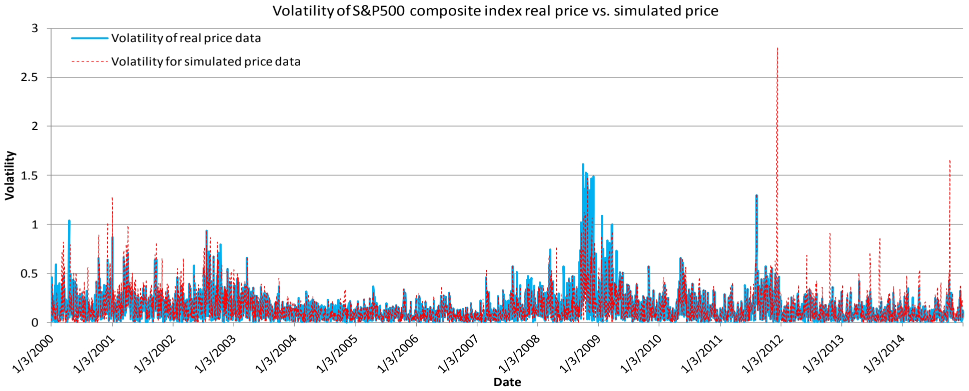

**Figure 5. S&P500 composite index volatility for real market price vs. simulated data**

*Source*: Own simulation of the multi-agent based model, forecast results obtained with Adaptive Modeler software application

## IV.Conclusion

The results of the academic studies of artificial stock markets using genetic programming are still in contradiction, mainly due to the variety of models which make it hard to classify and compare. Their complexity, lack of transparency and high number of degrees of freedom make understanding and further development very hard to achieve. Adaptive models focused on a single traded stock such as the one described in this paper represent a step further when it comes to stock market simulations techniques, stressing the importance of using an evolutionary model that evolves and adapts to the new market conditions.

The main improvements brought by this type of models are the following: the fitness function is computed as the actual return of the artificial agent after trading in the artificial stock market; there is no over-fitting the historical data during the learning process due to the fact that the model evolves in a time-incremental way, and it does not optimize on historical data. This model uses a high number of artificial agents, namely 2,000 agents and 2,000 continuously adapting trading rules, which increases model stability and reduces sensitivity to random factors. Trading signals given by the model are based on the interaction of all artificial agents, and not just on a single trading rule. The agent-based model is dynamic, constantly evolving and adapting to market conditions. Further research will focus on more simulations and statistical testing of the model accuracy and stylized facts.

## V.Acknowledgment

This work was financially supported through the project "Routes of academic excellence in doctoral and post-doctoral research - READ" co-financed through the European Social Fund, by Sectoral Operational Programme Human Resources Development 2007-2013, contract no POSDRU/159/1.5/S/137926.

## VI.References

1. Allen, F., & R. Karjalainen (1999), *Using genetic algorithms to find technical trading rules*, Journal of Financial Economics, 51, pp. 245–271.
2. Aloud, M., Fasli, M. & Tsang, E. (2013) *Modelling the High-Frequency FX Market: an Agent-Base Approach*, University of Essex, Colchester, PhD Thesis. http://fac.ksu.edu.sa/sites/default/files/MoniraAloud-Ph.D.Thesis.pdf , accessed October 24, 2014
3. Arthur, B., Holland, J., & LeBaron, B. (1997) *Asset Pricing Under Endogenous Expectations in an Artificial Stock Market*, Published In: Arthur, W.B., Durlauf, S.N., Lane, D. *The economy as an evolving, complex system II*, Redwood City, CA., Addison Wesley, pp. 15-44.
4. Demir, A., Shadmanov, A., Aydinli, C., & Eray, O. (2015). *Designing a forecast model for economic growth of Japan using competitive (hybrid ANN vs multiple regression) models*. Ecoforum Journal**,** 4(2), pp. 49-55**.** http://www.ecoforumjournal.ro/index.php/eco/article/view/174/111 , accessed June 18, 2015.
5. Holland, J., & J. Miller (1991), *Artificial adaptive agents in economic theory*, The American Economic Review, 81, pp. 365–370
6. Hommes, C. (2006), *Heterogeneous agent models in economics and finance*, in Handbook of Computational Economics, vol. 2, edited by L. Tesfatsion and K. L. Judd, chap. 23, pp. 1109–1186, Elsevier.
7. Koza, J. (1992), *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge.
8. LeBaron, B. (2006), *Agent-based computational finance*, in Handbook of Computational Economics, vol. 2, edited by L. Tesfatsion and K. L. Judd, 1 ed., chap. 24, pp. 1187–1233, Elsevier
9. Leigh, W., R. Purvis, & J. Ragusa (2002), *Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural networks, and genetic algorithm: a case study in romantic decision support*, Decision Support Systems, 32, pp. 361–377
10. Lux, T., , & S. Schornstein (2005), *Genetic learning as an explanation of stylized facts of foreign exchange markets*, Journal of Mathematical Economics, 41(1-2), pp. 169–196
11. Marchesi, M., S. Cincotti, S. Focardi, & M. Raberto (2000), *Development and testing of an artificial stock market*, in Modelli Dinamici in Economia e Finanza, Urbino.
12. Matsumoto, M., & Nishimura, T. (1998) *Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator*. ACM Trans. on Modeling and Computer Simulation, 8 (1), pp. 3-30.
14. Montana, D. (2002) *Strongly Typed Genetic Programming*, Evolutionary Computation, 3(2), pp. 199-230.
15. Phelps, S., Marcinkiewicz, M., & Parsons, S. (2005) *Using population-based search and evolutionary game theory to acquire better-response strategies for the double-auction market*. In Proceedings of IJCAI-05 Workshop on Trading Agent Design and Analysis (TADA-05).
16. Raberto, M., & S. Cincotti (2005), *Analysis and simulation of a double auction artificial financial market*, Physica A, 355, pp. 34–45.
17. Walia, V., Byde, A., & Cliff, D. (2003). *Evolving Market Design in Zero-Intelligence Trader Markets*. In IEEE International Conference on E-Commerce (IEEE-CEC03), Newport Beach, CA., USA.
18. Witkam, J. (2003). Altreva Adaptive Modeler simulation software, http://www.altreva.com/technology.html