12-2006

TrAgent: A Multi-Agent System for Stock Exchange

Raju Tatikunta
Southern Illinois University Carbondale

Shahram Rahimi
Southern Illinois University Carbondale, rahimi@cs.siu.edu

Pranav Shrestha
Southern Illinois University Carbondale

Johan Bjursel
George Mason University

Published in Tatikunta, R., Rahimi, S., Shrestha, P., & Bjursel, J. (2006). TrAgent: A multi-agent system for stock exchange. 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops, 505-509. doi: 10.1109/WI-IATW.2006.137

TrAgent: A Multi-Agent System for Stock Exchange

Raju Tatikunta[1], Shahram Rahimi[1], Pranav Shrestha[1], Johan Bjursel[2]

[1]*Department of Computer Science*
Southern Illinois University
Carbondale, IL 62901
[rahimi, rtati, pshres@cs.siu.edu]

[2] *School of Computational Sciences*
George Mason University
MS 5C3, Fairfax, VA 22030
cbjursel@gmu.edu

Abstract

TrAgent is a software-agent based model for a stock exchange such as the New York Stock Exchange and the procedure of trading securities on the trading floor. The model comprises the complete process from the initiation of a trading order until its execution. The software agent paradigm is the framework for implementing the proposed model. The properties of intelligent software agents meet the characteristics of the actors on the trading floor and provide capabilities for efficient distributed computing. Because of the space restriction, the main focus of this paper is on stock broker agent, one of the most important agents in TrAgent. Intelligence is provided to the stock broker agent in order to make reasoning and decisions on the profitability of a firm. This intelligence is provided using Fuzzy Expert System. The paper further discusses the design and development issues concerning different components of the model.

1. Introduction

Software agent technology is an exciting paradigm which can be efficiently applied to many problems related to Grid computing, particularly those that require dynamic behavior to reach a solution. Software agent technology has emerged as an enhancement of, if not an alternative to, traditional client/server model. Mobile agents can migrate to the desired remote peer and take advantage of local processing rather than relying on remote procedure calls (RPC) across the network. It could, therefore reduce the communication latency and bandwidth consumption in many applications with high data transfer rate.

Furthermore, agents are entities which function continuously and autonomously in a particular environment, and are able to carry out activities in a flexible and intelligent manner that is responsive to this dynamically changing environment. One such environment is the stock exchange market. Trading of stocks and bonds takes place almost every second following certain rules and regulations. As a result of this trading, large amount of diverse trade information is continuously exchanged between many different actors involved in the trade such as brokers, specialists and clerks. There are stock exchanges in all important financial centers of the world. This paper is based on the procedures carried out in the New York Stock Exchange (NYSE) [1].

The stock exchange model is efficiently replicated using mobile agent technology. The agents are designed to represent the actors involved in the stock exchange and perform tasks on their behalf. These mobile agents are capable of interacting with each other and with the external environment. They are also intelligent enough to make decisions based upon the ever-changing situations. TrAgent is a multi agent architecture encompassing the stock exchange in which components, their specifics, the interactions between components, and the intelligent components are defined and implemented.

This paper provides a concise overview of the system, while focusing mostly on the intelligence of the stock broker agent. The rest of the paper is organized as follows: section 2 briefly discusses TrAgent architecture and the agents involved in the model, while section 3 discusses the implementation of TrAgent architecture including the intelligent component involved. Finally, section 4 concludes the paper.

2. TrAgent Model– A Distributed Trading Architecture

TrAgent is a novel distributed architecture encompassing the stock exchange [4]. In this section, the components of TrAgent, their specifications, and the interactions between components and their intelligence through a case analysis are defined.

The different components involved in the process are characterized with the aim of precisely defining their attributes, functionality, interactions, and responsibilities. Figure 1 provides an overview of the trading floor and its components. To better observe the actors relationships and understand their functionality, a simple trading scenario, where a transaction takes place, is set up as follows (Note that some details are left out intentionally to simplify the scenario):

1. The investor places an order with a broker.
2. The broker sends the order to the firm's clerk on the floor of the exchange.
3. The clerk gives the order to the firm's floor broker .

Proceedings of the 2006 IEEE/WIC/ACM International Conference on
Web Intelligence and Intelligent Agent Technology (WI-IAT 2006 Workshops)(WI-IATW'06)
0-7695-2749-3/06 $20.00 © 2006 IEEE

4. The floor broker goes to the specialist's post where the stock is traded and begins negotiating with the trading crowd .
5. The broker agrees on a price.
6. The order is executed.
7. The floor broker reports the trade to the clerk.
8. The clerk confirms the order with the broker.
9. The broker confirms the trade with the investor.

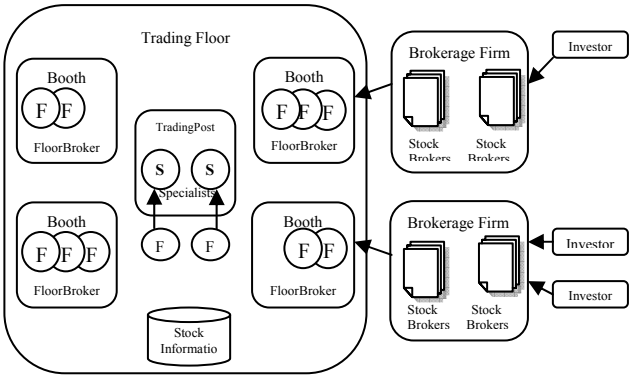

Figure 1. Trading floor and its components

2.1 Investor Agent

The investor agent is not a part of the actual exchange. However, it is critical as the initiator of any transaction. This agent is provided with a list of firms it is interested in by the stock broker agents, ranked according to the profitability. Upon deciding on the firm the Investor agent requests the stock broker agent to initiate the trade on its behalf. The Investor agent can also have access to external information such as stock information and the latest quotes. When an order has been executed, the stock broker agent reports the result to the Investor agent. Hence, the Investor agent has to furnish methods interfacing with the brokers.

2.2 Stock Broker Agents

A broker is a mobile agent that acts as an intermediary between the buyers and sellers in trading securities, commodities, and other stocks. The stock broker agent manages the transactions for the investor agent as well as provides analytical functions. Moreover, it ensures that investor agent orders are executed and relayed to the floor broker agent that executes the orders.

An investor agent may require two services from the stockbroker: order a transaction or stock analysis. In the latter case, the stockbroker agent performs the analysis, returns the result and goes back to being idle. In the former case, the stockbroker agent validates the order. Thereafter, a booth on the trading floor is contacted in order to access a floor broker agent. After the order is delegated to the floor broker agent, the stock broker agent returns to its idle state. When a transaction has been executed, the stock broker agent receives the execution report, summarizes the result, and informs the investor.

While performing stock analysis, the stock broker agent does not predict any thing related to the future values of the firm, rather it assists the investor agent by providing it with an ordered list of firms in which the investor agent is interested. The firms in this list are ranked based on their profitability. It is up to the investor to decide whether it is willing to sell or invest on a firm looking at the profitability ratios. To explain it further, the stock broker agent uses its intelligence to decide whether the profitability of a firm is low, medium, high or very high, based on some profitability ratios. The intelligent component is developed using the fuzzy expert system in this model. The details of the implementation of the fuzzy expert system are discussed in section 3.3.

2.3 Floor Broker Agent

The workspaces around the perimeter of the trading floor where member firms and independent floor brokers receive orders are referred to as Booths. Each Booth includes one or more floor brokers.

A floor broker agent will stay idle until an order is received from the stock broker. The component must provide functionality for a stock broker agent to place orders, and also for other floor broker agents and specialist agents to negotiate trade. Since a floor broker agent is at the booth and stocks are traded at trading posts, the floor broker agent has to find the correct trading posts upon receiving an order and then move to it. After arriving at the post, a specialist agent is contacted. The floor broker agent then places the bid with the specialist agent and anticipates a response. If the bid is accepted, the floor broker agent prepares to move back to the booth and report the trade information to the stock broker agent. However, if that is not the case, the floor broker agent may place a new bid or wait for a better offer to come. The floor broker cannot leave the post unless the order is executed.

2.4 Specialist Agent

A specialist agent has many roles and responsibilities, and therefore, only the functionality of trading one-on-one with a floor broker agent is detailed here. In order to trade, the specialist agent provides methods for negotiating prices with the floor broker agent. Functions for requesting trade information (e.g., price and quantity) are furnished as well. The specialist agent remains idle until a floor broker agent actually places a bid. If the specialist agent accepts the offer, it prepares the trade and reports the result to the floor broker agent. If the bid is rejected, the specialist agent waits for a new bid from the floor broker agent or records the order and attempts to match it with new incoming orders. It also interacts with the floor official, as

well as the trading information systems (e.g., SuperDot) to store and receive necessary information.

3. Implementation of TrAgent

This section briefly describes the implementation particulars of the TrAgent model. TrAgent is implemented in JADE (Java Agent Development Environment) [3] and its design comprises two major components:
a) The software agents and their communication protocols [12] and environments.
b) The intelligence and reasoning for the agents, required for taking actions.

The whole system is developed as a platform and contains containers (also called agencies) for every group of entities involved in the model. All the containers register themselves with the platform upon creation. The main container holds the Agent Management System (AMS) agent and the Directory Facilitator (DF) agent. AMS agent is the authoritative body of the platform which provides the naming service for all the agents residing in the platform. Equally important is the role of DF agent which provides a Yellow Pages service by which an agent can locate other agents providing the services it may require. Figure 2 illustrates the outline of the TrAgent architecture in JADE.

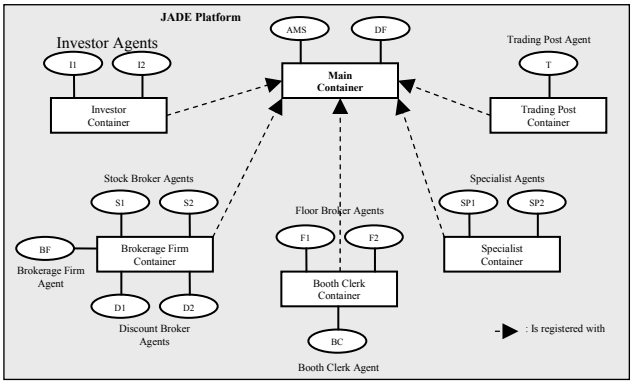

Figure 2. TrAgent Model in JADE

3.1 Investors Agents

Investor agents are created in the investor container. Every investor agent uses the DF service to find a brokerage firm agent which provides the investor with a stock broker agent ID (name and location) to perform the required transactions. The TrAgent model only allows the Investor agents to perform trade, i.e., buy/sell stock or ask for analysis.

3.2 Brokerage Firm and Stock Broker Agents

Brokerage firm and stock brokers are created in the brokerage firm container. The brokerage firm agent publishes its services with the DF agent so investor agents can locate it for any needed services. It provides for requests from investor agents by assigning an available stock broker agent to each request. The brokerage firm maintains a list of all active stock brokers.

Stock brokers register only with one brokerage firm agent. These agents also provide the brokerage firm with their status information, which can be either *idle* or *busy*. A stock broker agent manages the transactions for the investor agent and provides analysis of the firms, upon request. For analyzing the stock, the stock broker agent utilizes its fuzzy expert system to provide its advice. Stock broker agents need the service of a floor broker agent to perform the actual stock transaction. They use the DF agent to find a booth clerk agent so they can request for a floor broker agent.

3.3 Intelligence for Stock Broker Agent

The intelligence component that has been applied to stock broker agent is a Fuzzy Expert System (FES). This kind of systems constitute an extension of the classical Rule-Based System since they deal with fuzzy rules instead of the classical logical rules [7], and have been successfully applied to a wide range of problems dealing with uncertainty.

In our approach, a stock broker agent is the entity that provides the investor agent with a ranked list of the firms in which the investor agent is interested in, ranking them according to their possible profitability. As described earlier, to rank the firms the stock broker agent utilizes its FES engine. However, to do so, the FES receives inputs from stock brokers who belong to the firm. Detailed discussion is presented in ensuing sub-sections.

3.3.1 Membership Functions

The profitability of a firm depends on various criteria [10] [11]. In TrAgent four major profitability factors that influence the profit of a firm have been considered. These factors are return on equity (ROE), return on assets (ROE), operational profit margin (OPM), and net profit margin (NPM). The stock broker agent collects the current values of these factors and then decides on the profitability of the firms using its FES. It then generates an ordered list of the firms according to their profitability and provide that to the investor agent.

These four profitability factors are the input variables to the Fuzzy expert system. They are fuzzified into four possible fuzzy sets: not satisfactory (NSAT), medium (MED), satisfactory (SAT) and very satisfactory (VSAT). Depending upon their membership values each parameter can exist in more than one fuzzy set.

The membership functions for the input and output variables, to and from the FES, are of triangular type. However, deciding on the ranges of the fuzzy sets for the

Proceedings of the 2006 IEEE/WIC/ACM International Conference on
Web Intelligence and Intelligent Agent Technology (WI-IAT 2006 Workshops)(WI-IATW'06)
0-7695-2749-3/06 $20.00 © 2006 IEEE

input variables is a critical task. These ranges vary for different industries in different sectors, and it is almost impossible to come up with general ranges that work for multiple industries. The Fuzzy expert systems in TrAgent have been knowledge engineered using the experts in the field who provided the initial ranges for the input and output fuzzy sets and helped in tuning and adjusting the fuzzy sets and the fuzzy rules. To do so the expert considered the statistical information of the firms in a particular industry and came up with the ranges particular to that industry. One set of such ranges for the membership functions, which are provided for the "business software and services" industry in the "technology" sector, are given in Table 1.

Table 1. Showing the input fuzzy sets with their ranges

Return on Equity A1:		**Return on Assets A2:**	
NSAT	A1 < 3%	NSAT	A2 < 6%
MED	1 < A1 < 9%	MED	4 < A2 < 10%
SAT	8 < A1 < 16%	SAT	8 < A2 < 15%
VSAT	A1 > 14%	VSAT	A2 > 13%
Operational Profit Margin A3:		**Net Profit Margin A4:**	
NSAT	A3 < 6%	NSAT	A4 < 5%
MED	4 < A3 < 11%	MED	3 < A4 < 12%
SAT	8 < A3 < 16%	SAT	9 < A4 < 18%
VSAT	A3 > 13%	VSAT	A4 > 15%

3.3.2 Fuzzification and Aggregation of the Activated Rules

The FES used by the stock broker agent in our model is a three stage process. In the first stage, the input variables are fuzzified. This comprises of the process of transforming the crisp values to grades of membership for linguistic terms of fuzzy sets. The membership functions are used to associate grades to each linguistic term. The second stage is the decision making logic where the rules that get activated based on the input values are combined. The third stage provides the final output, a fuzzy set, after combining the rules that are fired by the FES. This fuzzy set is defuzzified in order to obtain a crisp value in the last stage. Defuzzification is described in section 3.3.3. The stock broker agent decides on the profitability of a firm based upon this crisp value.

Fuzzy rules are generally generated heuristically. In every FES there exists an average of 85 rules. As an example and to represent the general format of the rules in the rule base of the FES, here is a sample rule:

If *ROE* is VSAT **AND** *ROA* is VSAT **AND** *OPM* is SAT **AND** *NPM* is SAT **THEN** Profitability is HIGH

Where ROE, ROA, OPM, and NPM are the linguistic variables representing the input to the fuzzy system and profitability is the system output and NSAT, MED, SAT, SAT are the corresponding fuzzy sets which represent the linguistic terms being associated with a membership value of μ that specifies the degree to which the given input variable satisfies the output. The number of the rules in an expert system could increases exponentially with the dimensionality of the input space. However, one of the benefits of using a fuzzy expert system is to simplify the environment and therefore control the size of the rule base. During the process of the development of the rule-base attention has been given to remove contradictory rules and the rules that may not make sense from the point of view of the experts. For instance, if the NPM of a firm is VSAT the OPM cannot be NSAT. Rules with such contradictory conditions have been eliminated. Since each fuzzy if-then rule is linguistically interpretable, rules can easily be understood and filtered by human experts.

Based upon the values for the input variables, and depending on the fuzzy sets these input variables belong to, the rules whose antecedents are matched are fired by the FES. The aggregation of the rules that are fired is a two stage procedure in our model. Initially the set of the rules fired are combined. This is done by using the Mamdani's inference method [9] where the minimum membership value of the antecedents is the degree of truth. The output membership function is then clipped off at a height corresponding to the rule antecedents computed degree of truth. The output is still fuzzy in nature. As an example let us consider the values for return on equity, return on assets, operational profit margin and net profit margin to be 4%, 8%, 15% and 18% respectively. The process of fuzzification and aggregation using these sample values is shown in Table 2 where the output is computed.

Table 2. Showing inputs and outputs of the rules fired

Input				**Output**			
ROE	**ROA**	**OPM**	**NPM**	**LOW**	**MED**	**HIGH**	**VHIGH**
4%	8%	15%	18%	0.00	0.406	0.00	0.00
				0.00	0.00	0.312	0.00
				0.00	0.132	0.00	0.00
				0.00	0.00	0.115	0.00
Output:				**0.406/MED + 0.312/HIGH**			

3.3.3 Defuzzification

The result of combining the fired rules is an aggregated output fuzzy set. For most applications such as this one, there is a need for a single action or a "crisp" output, to emanate from the inferencing process. This will involve the defuzzification of the solution set. Defuzzification is a process of selecting a crisp number that represents the membership function. The final output obtained in the example in table 2 is:

{0.406/MED, 0.312/HIGH}.

This is a fuzzy set and need to be defuzzified in order to get the crisp output. There are various techniques available for defuzzification. The center of area (COA) method is

used in the proposed system. The universe of discourse for the output variable is the profitability, and the profitability made for this example, using COA method, is 47%. Figure 3 illustrate the defuzzification task for the given example.

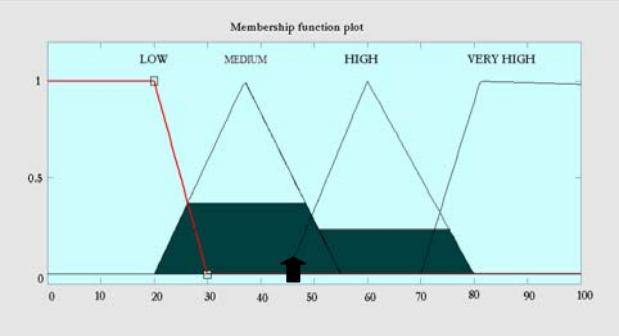

Figure 3. Defuzzificatioin of the output for table 2, using center of area method

3.4 Booth Clerk and Floor Brokers Agents

The booth clerk agent and its floor broker agents are created inside the booth clerk container. The booth clerk agent also publishes its services with the DF agent. It serves requests from stock broker agents for available floor broker agents. Similar to stock broker agents, floor broker agents also register only with a single booth clerk agent, which also provide their status information, *idle* or *busy*. A floor broker agent stays idle until an order is received. Upon receiving a buy order, it locates the corresponding specialist agent and executes the trade. After a transaction is completed, the floor broker reports the result of the trade to the stock broker agent.

3.5 Specialists and Trading Post Agents

The specialist agents and the trading post agent are created in separate containers. The trading post agent is solely responsible for providing stock information to the specialist agent and also to the stock broker agents. The trading post agent maintains the stock database and publishes its services with the DF agent when it starts execution.

Similarly, specialist agents register with the DF agent so that the floor broker agents can locate them. They receive requests from the floor broker agents for trading stocks, carry out the execution and report the results back to the floor broker agents.

4. Conclusion

This paper presented a model encompassing the NY stock exchange from when an investor issues a trade order until the transaction is executed and reported. The majority of the recent literature suggests the use of mobile agents and artificial intelligence as a tool for surveying the market and decision support. However, no system known to the authors models the actual stock market as of this day. The aim of this work is to provide a model for trading which can be used as a platform to study the flow of trading, to implement trading strategies, to study possible future stock exchange technology and finally to provide a platform for total automation of stock exchange.

5. Reference

[1] *New York Stock Exchange*, Retrieved July 31, 2006 from http://www.nyse.com

[2] W. J. Eiteman, C. A. Dice, and D.K. Eiteman, *The Stock Market*, McGraw-Hill Inc., Fourth Edition.

[3] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, *JADE: A White Paper*, http://jade.tilab.com/

[4] J. Bjursell, S. Rahimi, and C.F. Wang, "Modeling the Stock Exchange Market Using Mobile Agent Technology and its Security Issues," *Intl. Journal of Education and Information Technology*, Vol. 1, No. 1, September 2004, pp. 28-38.

[5] L. Nedovic, and V. Devedzic, "Expert System in Finance – A Cross-section of the field", *Expert Systems With Applications*, Vol.23, No.1, 2002, pp. 49-66.

[6] L. A. Zadeh, "Fuzzy sets", *Information and Control,* vol. 8, 1965, pp. 328-353.

[7] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decisions processes", *IEEE Transactions on Sytems, Man, and Cybernetics,* vol. 3, 1973, pp. 28-44.

[8] N. Suri, J.M. Bradshaw, M.R. Breedy, P.T. Groth, G.A. Hill, and R. Jeffers, "Strong mobility and fine-grained resources control in NOMANDS". *Proc. of 2nd Intl. Symposium on Agents Systems and Applications and the 4th Intl. Symposium on Mobile Agents*, vol 1882 of LNCS, 2000, pp. 2-15

[9] E.H. Mamdani, and S. Assilian, "An experiment in linguistics synthesis with a fuzzy logic controller", *Intl. journal of Man-Machine Studies,* 7(1), 1975, pp. 1-13.

[10] C. Zopounidis, M. Doumpos, and N.F. Matsatsinis, "Application of the fineva multicriteria knowledge-based decision support system to the assessment of corporate failure risk", *Foundation of Computing and Decision Sciences,* vol. 21, 1996, pp. 233-251.

[11] S. C. Gold, and P. Lebowitz, "Computerized stock screening rules for portfolio selection", *Financial Services Review,* vol. 8, 1999, pp. 61-70.

[12] The Foundation for Intelligent Physical Agents, Retrieved July 31, 2006 from http://www.fipa.org