Development of An Agent-Based Framework for Stock Market Trading

O.E. OYETUNJI[1], K.G. AKINTOLA[2]

[1] *Department of Computer Science, The Federal University of Technology, Akure*

[2] *Department of Software Engineering, The Federal University of Technology, Akure*

Abstract- Trading online demands time bandwidth and adequate knowledge. Online stock trading is a system that allows buy and selling of stocks through the Internet. This paper presents a development of an Agent-based Framework for stock market trading. The system allows the use of agents in stock trading on behalf of traders. The system includes creating investor's agents which are: market maker agent, informed agent, uninformed agent and the hold agent. Initial capital are allocated to the investor's agents, the agents now use the capital to trade on behalf of the investors and making decision to meet the user's investment goals which include optimum return on investment. The key issues addressed include assembly and incorporating diverse information sources with trading agents, and providing decision-making for investor's agent in the stock market using trading indicator for informed agent. The prototype of the system has been implemented and found to be very efficient in stock trading based on return on investment.

Indexed Terms- Stock, Multi-agent, JADE, Investor, Market

I. INTRODUCTION

In modern society, information and access is getting more and more important. During the last couple of years, there is a strong tendency towards mobility. This implies that there is an increasing need for being online and having access to information all the time. This trend is also present in the world of stock trading. People want to use their mobile phones, PDA or laptops to do online shopping. These devices cannot be online all the time (without cost) which can be a limitation for some stock trading applications. Fortunately, the concept of agents can potentially solve the aforementioned problem. Agents are software programs that can travel autonomously from host to host to perform one or more tasks on behalf of users. They can communicate (and even negotiate) with other agents. The user can go offline after the agent is sent out and can come online again when the agent has performed his tasks. This is very useful for situations in which the user cannot stay online for a long time.

Manvi and Ventararam (2004) remark that the term 'agent' used in its pure and unqualified form, denotes a static software entity with a well-defined role, typically acting on behalf of a human or another software component, which may be used in a variety of applications. Agent acts autonomously on behalf of users with the ability to precisely initiate migration by itself and execute its owner's request and move independently from one host to another in order to resume execution (Byun et al., 2005).

An agent can interact on and ask/receive paradigm with a static agent residing on the host that offers the information needed to perform an operation successfully. Thus, the agent can find out only the relevant information or data and avoid unnecessary network traffic. In this scenario of information gathering in the Internet, commercial aspects (Kalakota and Whinston, 1996) have become more and more relevant, e.g., the agent has to pay for information, or in other possible application scenarios the agent will be paid for a service it offers. As a consequence, agents have to be equipped with electronic commerce capabilities to be an autonomous entity in the electronic market place. This fortunately demand an agent-based trading model that uses a behavioral finance approach in which the agent exhibit a bias in decision- making process based on pre-set rules and without limitations.

A stock market is a similar designated market for trading various kinds of securities in a controlled, secure and managed the environment. Stock markets provide a secure and regulated environment where market participants can transact in shares and other eligible financial instruments with confidence with zero to low operational risk. Operating under the defined rules as stated by the regulator, the stock markets act as primary markets and as secondary markets. The primary objective of investing is to ensure that every person is able to meet his or her future financial objectives. Rise in inflation makes it inadequate for individuals to simply earn and save some part of their incomes. To meet the price increases due to inflation, investments become important. The stock market is one of the oldest and most popular investment avenues due to several benefits of investing in stocks. Stock investment offers plenty of benefits which includes: higher liquidity, versatility, acquiring ownership and right to vote, regulatory Environment and Framework.

The agent paradigm has the following features (Al-Kasassbeh and Adda,2007):

a. Autonomy: an agent operates without intervention;
b. Social Ability: agents interact with each other through agent communication language;
c. Goal Driven: agents exhibit goal-directed behavior; and
d. Reactivity: agents perceive and respond to their environment.

II. RELATED WORK

In this section, an attempt is made to review some specific research works on agents with a view to identifying their motives, objectives, methodologies, their contribution to knowledge and their limitations.

A research titled “Agents principles of operation” was presented in (Yariv et al., 1988) whose motivation was because of several reasons why agents (termed masters) would like to create other agents (termed slaves) and delegate tasks to them. One is performance. A master agent can continue to perform other tasks in parallel with the slave agent. To create an agent application that is weary of inventing and re-inventing solutions to recurrent problems, and give an agent design patterns can will help in capturing solutions to common problems in agent design. The methodology involved using Aglet for developing an agent. During the early work on Aglets the developers recognized a number of recurrent patterns in the design of agent applications. Several of these patterns were given intuitive meaningful names such as Master- Slave, Messenger, and Notifier. They were implemented in Java and included in the first release of the Aglets Workbench. These early patterns were found to be highly successfully for jump-starting users who were new to Aglets and the agent paradigm. The research is only theoretical and lacking in practical design and implementation.

Oyetunji and Akintola (2019) developed a mobile agent platform for buying and selling of books online. The proposed system architecture can address the problem of expensive connection with limited bandwidth. After launching the mobile agent’s seller/buyer can disconnect from mobile devices later and later on can reconnect to devices for results. The proposed system is demonstrated by creating a buyer agent and seller agent using JADE (Java Application Development Framework) in NetBeans Integrated Development Environment. The experimental demonstration of the agents shows that the adoption of mobile agents for trading is feasible. In this system the decision to buy a book was only based on the prices of the books, no technical indicators were used.

III. METHODOLOGY

This research development started from reviewing of relevant textbooks, journals, magazines, conferences and workshop papers on agent for stock trading. It also involves the development of an Agent-based Framework which makes use of four types of agents namely: market -maker agent, informed agent, uninformed agent and hold agent. The market-marker agent will be in charge of the smooth coordination of the stock trading between the other three agents. Two standard indicators will be available to be used by the informed agents to help in deciding whether to sell, buy or hold the stock. The uniformed agent has the capacity to make transactions illogically since such agent will not be aware of the indicators that can help in making sell, buy or hold decisions. The hold agent buys stocks, keep for some period and later sell it.

The model will be implemented using JADE (Java Agent Development Framework) to develop the multi-agent system and MySQL will be used for the data management of the system. The model will be evaluated using profitability metrics. The system architecture is presented in Figure 1.

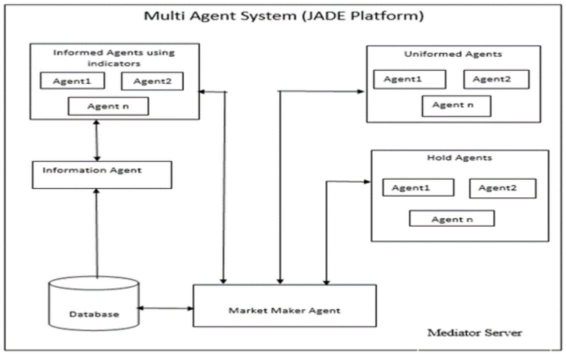


Figure 1: System Architecture

The Agent-based Framework will be used to develop five types of agents which are:

a. market -maker agent,

b. informed agent,

c. uninformed agent,

d. hold agent and

e. Information agent

The market-marker agent as shown in figure 3.1 will be in charge of the smooth coordination of the stock trading between the other three agents. The market-maker sets bid and ask prices (P_b and P_a respectively) at which it is willing to buy or sell one unit of the stock at each time. The bid and ask prices at time period *i* are denoted as p_b^i and p_a^i . The stock has a true underlying value (or fundamental value) V^I at time period *i*. All market makers are informed of the fundamental values of shares. At time period *i*, a trader is selected and allowed to place either a (market) buy or (market) sell order for some quantity of stock. The informed agent makes use of indicators to determine the safe period to buy, sell or hold stocks. The uniformed agent have the capacity to make transactions illogically since such agent will not be aware of the indicators that can help in making sell, buy or hold decisions. An uninformed agent trader will place a buy or sell order for one unit with equal probability, or no order with some probability if selected to trade and the hold agent buys stocks, keep for some period and then sell it.

The two standard indicators which will be available to be used by the informed agents to help in deciding whether to sell, buy or hold the stock are the following:

a. Simple moving Average Indicator

An informed trader using simple moving average (SMA) indicator will place a buy order when the shorter-term SMA crosses above the longer-term SMA but will place a sell order when the shorter-term SMA crosses below the longer-term SMA, else will place a hold order. Short-term SMA means addition of the closing prices of the stock for a small number of time periods for example 7 days while Long-term SMA means addition of the closing prices of the security for a long number of time periods for example 21days.

The SMA is calculated as follows:

$$SMA = \sum_{i=1}^{n} \frac{x_i}{n} \qquad (1)$$

where x_i is the sum of the prices and n is the number of total periods

b. Price Rate of Change (ROC) Indicator

An informed agent trader using Price Rate of Change (ROC) indicator will place a buy when the ROC valve is less than zero and will place a sell order when the ROC value is greater than zero, else will place a hold order. ROC is calculated as follows:

$$\text{ROC} = \left(\frac{\text{C} - \text{C}p}{\text{C}p}\right) * 100 \qquad (2)$$

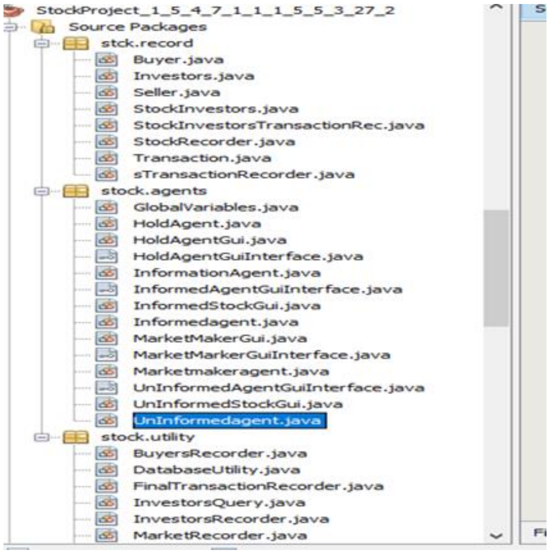
where C is the closing price of most recent period, Cp is the closing price *n* periods before most recent period.

IV. SYSTEM IMPLEMENTATION AND EVALUATION

The implementation techniques adopted in the development of the agent. An Agent-based Framework was developed producing four types of agents which are: market -maker agent, informed agent, uninformed agent and hold agent. The market-marker agent will be in charge of the smooth coordination of the stock trading between the other three agents. Two standard indicators will be available to be used by the informed agents to help in deciding whether to sell, buy or hold the stock. The uniformed agent have the capacity to make transactions illogically since such agent will not be aware of the indicators that can help in making sell, buy or hold decisions. The hold agent buys stocks, keep for some period and then sell it. The model will be implemented using JADE (Java Agent Development Framework) to develop the multi-agent system and MySQL will be used for the data management of the system. The model will be evaluated using profitability metrics.

4.1 IMPLEMENTING THE AGENT DESIGN

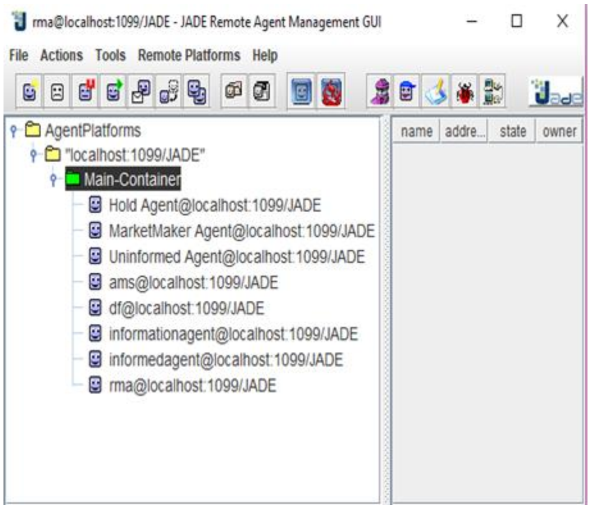
In the agent implementation, a modular implementation approach incorporating nine Java classes and four interface interacting among themselves was adopted. The classes are: HoldAgent, HoldAgentGuiInterface, InformedAgent, InformedAgentGuiInterface, InformationAgent, MarketMarkerGuiInterface, Marketmakeragent, UnInformedAgentGuiInterface, UnInformedagent while the interfaces are: HoldAgentGui, InformedStockGui, MarketMakerGui, UnInformedStockGui in the JADE platform as shown in Figure. 2



```
StockProject_1_5_4_7_1_1_1_5_5_3_27_2
  Source Packages
    stck.record
      Buyer.java
      Investors.java
      Seller.java
      StockInvestors.java
      StockInvestorsTransactionRec.java
      StockRecorder.java
      Transaction.java
      sTransactionRecorder.java
    stock.agents
      GlobalVariables.java
      HoldAgent.java
      HoldAgentGui.java
      HoldAgentGuiInterface.java
      InformationAgent.java
      InformedAgentGuiInterface.java
      InformedStockGui.java
      Informedagent.java
      MarketMakerGui.java
      MarketMarkerGuiInterface.java
      Marketmakeragent.java
      UnInformedAgentGuiInterface.java
      UnInformedStockGui.java
      UnInformedagent.java
    stock.utility
      BuyersRecorder.java
      DatabaseUtility.java
      FinalTransactionRecorder.java
      InvestorsQuery.java
      InvestorsRecorder.java
      MarketRecorder.java
```

Figure 2 : Project Java Classes

The GUI agents provide framework for visually controlling and monitoring the agents as they move from host to host as shown in Figure 3.



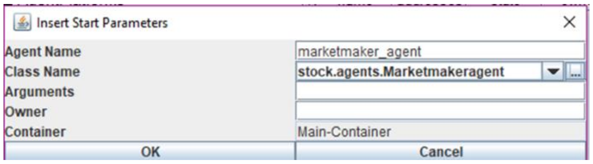
rma@localhost:1099/JADE - JADE Remote Agent Management GUI

```
AgentPlatforms
  "localhost:1099/JADE"
    Main-Container
      Hold Agent@localhost:1099/JADE
      MarketMaker Agent@localhost:1099/JADE
      Uninformed Agent@localhost:1099/JADE
      ams@localhost:1099/JADE
      df@localhost:1099/JADE
      informationagent@localhost:1099/JADE
      informedagent@localhost:1099/JADE
      rma@localhost:1099/JADE
```

Figure 3: JADE Remote Agent Management Gui

The MarketMaker Agent Functions

The MarketMaker Agent (Stock_Trading package MarketMakeragent class), also have a graphical interface . Through this interface the marketmaker coordinate the trading between the other trading agent which are informed agent, uninformed Agent and hold agent that are trading on their invenstor's behalf, as shown in Figure 4



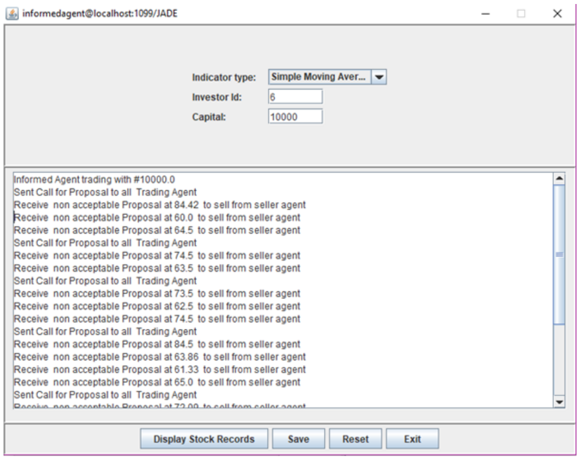
Insert Start Parameters

Agent Name	marketmaker_agent
Class Name	stock.agents.Marketmakeragent
Arguments	
Owner	
Container	Main-Container

Figure 4: MarketMaker Agent form Interface

The Informed Agent Functions

The Informed Agent (Stock_Trading package Informedagent class), also have a graphical interface . Through this interface the the investor's ID and the capital which the investor is willing to trade with is given to the agent and the indicator to be used for trading is specifed for trading to commence on invenstor's behalf, as shown in figure 4. The informed agent gets the stock prices from information agent who get the prices from marketmaker database. When the prices are received indicator(s) determine whether it is save to sell or hold or buy. Using a simple moving average Indicator the save period is determine using the trading graph as shown in figure 5. When sma21(the blue line) is above sma7 (the red line), the agent(s) buy stock because it is a save period to buy stock(s) and when sma7 (the red line) is above sma21(the blue line) the agent(s) sell stock(s) because it is a save period to sell stocks else the stock is on hold by the agent(s) At this point proposal is sent through Call for Proposal to other trading agent and when it is accepted, a replied is sent back and the reply is accepted by the informed agent. When it is an acceptable proposal and at that price, trading occcurs(either to buy or sell) but rejected when it is a non acceptable proposal(the agent holds trading activities till a save period is reached).



Indicator type: Simple Moving Aver...
Investor Id: 6
Capital: 10000

```
Informed Agent trading with #10000.0
Sent Call for Proposal to all  Trading Agent
Receive  non acceptable Proposal at 84.42  to sell from seller agent
Receive  non acceptable Proposal at 60.0  to sell from seller agent
Receive  non acceptable Proposal at 64.5  to sell from seller agent
Sent Call for Proposal to all  Trading Agent
Receive  non acceptable Proposal at 74.5  to sell from seller agent
Receive  non acceptable Proposal at 63.5  to sell from seller agent
Sent Call for Proposal to all  Trading Agent
Receive  non acceptable Proposal at 73.5  to sell from seller agent
Receive  non acceptable Proposal at 62.5  to sell from seller agent
Receive  non acceptable Proposal at 74.5  to sell from seller agent
Sent Call for Proposal to all  Trading Agent
Receive  non acceptable Proposal at 84.5  to sell from seller agent
Receive  non acceptable Proposal at 63.86  to sell from seller agent
Receive  non acceptable Proposal at 61.33  to sell from seller agent
Receive  non acceptable Proposal at 65.0  to sell from seller agent
Sent Call for Proposal to all  Trading Agent
Receive  non acceptable Proposal at 72.00  to sell from seller agent
```

Figure 5: Informed Agent Interface

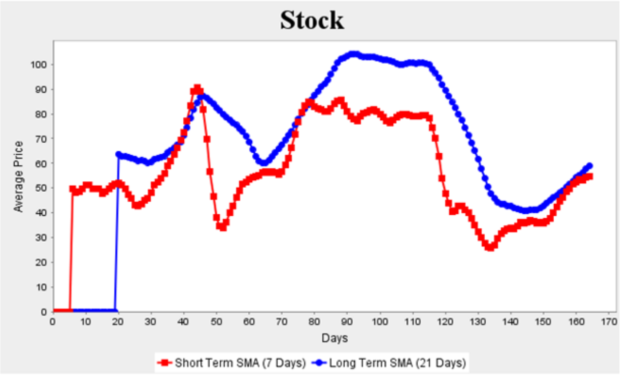
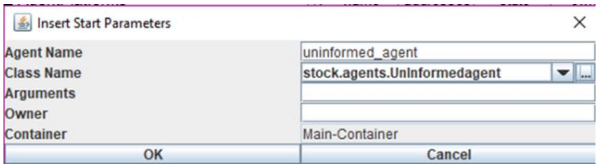


Figure 6: Trading Graph

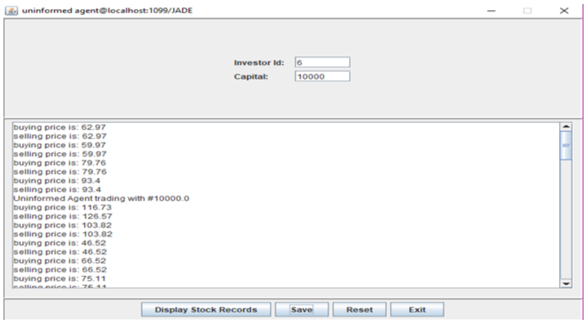
The Uninformed Agent Functions

The Uninformed Agent (Stock_Trading package UnInformedagent class) is created by the investor as shown in Figure 6 .Through the graphical interface through which the investor's ID and the capital with which the investor is willing to trade is given to the agent and the trading commence, as shown and figure 7. The Uninformed agent trades irrationaly depending on the current trading price when the agent enter the market on the investor's bahalf. The uninformed agent does not know the save time to buy or sell . The uninformed agent takes the investor's capital and trade.



Agent Name	uninformed_agent
Class Name	stock.agents.UnInformedagent
Arguments	
Owner	
Container	Main-Container

Figure 7: Creating an uninformed Agent



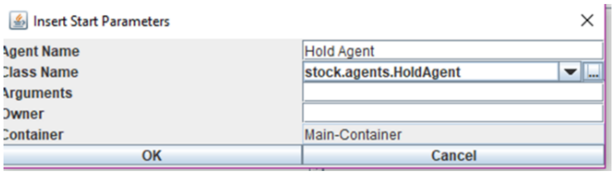
Investor Id: 6
Capital: 10000

```
buying price is: 62.97
selling price is: 62.97
buying price is: 59.97
selling price is: 59.97
buying price is: 79.76
selling price is: 79.76
buying price is: 93.4
selling price is: 93.4
Uninformed Agent trading with #10000.0
buying price is: 116.73
selling price is: 126.57
buying price is: 103.82
selling price is: 103.82
buying price is: 46.52
selling price is: 46.52
buying price is: 66.52
selling price is: 66.52
buying price is: 75.11
selling price is: 75.11
```

Figure 8: Uninformed Agent Interface

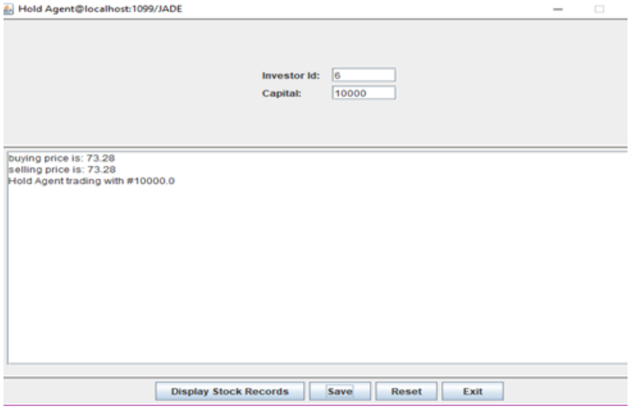
The Hold Agent Functions

The Hold Agent (Stock_Trading package HoldAgent class) is created by the investor as shown in figure 8. Through the graphical interface the investor's ID and the capital with which the investor is willing to trade is given to the agent and the trading commence after sometimes when the agent has hold the capital , as shown and figure 9. The Hold agent trades on the investor's bahalf after a while depending on the current trading price despite holding the investor's capital for sometimes. The hold agent does not know the save time to buy or sell and figure 10 shows the status list of the Trading Agents.



Agent Name	Hold Agent
Class Name	stock.agents.HoldAgent
Arguments	
Owner	
Container	Main-Container

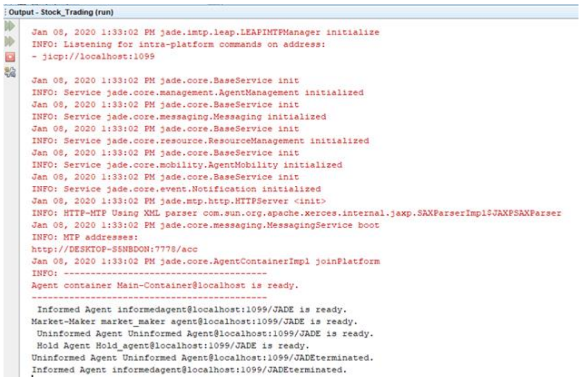
Figure 9: Creating an Hold Agent



Investor Id: 6
Capital: 10000

```
buying price is: 73.28
selling price is: 73.28
Hold Agent trading with #10000.0
```

Figure 10: Hold Agent Interface



```
Jan 08, 2020 1:33:02 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://localhost:1099

Jan 08, 2020 1:33:02 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Jan 08, 2020 1:33:02 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Jan 08, 2020 1:33:02 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
Jan 08, 2020 1:33:02 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Jan 08, 2020 1:33:02 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Jan 08, 2020 1:33:02 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
Jan 08, 2020 1:33:02 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://DESKTOP-S5NBDON:7778/acc
Jan 08, 2020 1:33:02 PM jade.core.AgentContainerImpl joinPlatform
INFO: --------------------------------------
Agent container Main-Container@localhost is ready.
--------------------------------------------
 Informed Agent informedagent@localhost:1099/JADE is ready.
Market-Maker market_maker agent@localhost:1099/JADE is ready.
 Uninformed Agent Uninformed Agent@localhost:1099/JADE is ready.
 Hold Agent Hold_agent@localhost:1099/JADE is ready.
Uninformed Agent Uninformed Agent@localhost:1099/JADEterminated.
Informed Agent informedagent@localhost:1099/JADEterminated.
```

Figure 11: Status list of the Trading Agents

SYSTEM EVALUATION

In this research work, two criteria are used to create a successful book trading agent. They are: market maker agent, informed agent, uninformed agent, hold agent; and the JADE platform is used in implementing the communication between them. Users of the system find it amazing to invest in stock trading that they want in no time and with less stress and with great return of investment especially when informed agent is used as shown in figure 11

Trading result for Informed Investor's using #10,000 as capital

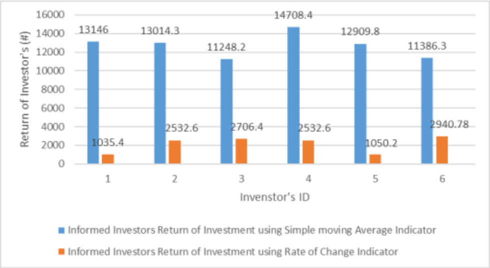


Figure 12: Informed Investor's Return of Investment using #10,000 as capital

Six informed agent investors were given #10,000 each to trade using simple moving average indicator and rate of change indicator. Using simple moving average indicator it was observed that more than the invested capital was made as return of investment as shown in figure 12 after which the informed investor's agents has bought stock during the save period to buy stock (when sma21 is greater than sma7) and sold their stocks during the save period to sell their stock(when sma7 is greater than sma21) and little return of investment was made using Rate of change using Rate of Change indicator.

Trading result for Uninformed Investors

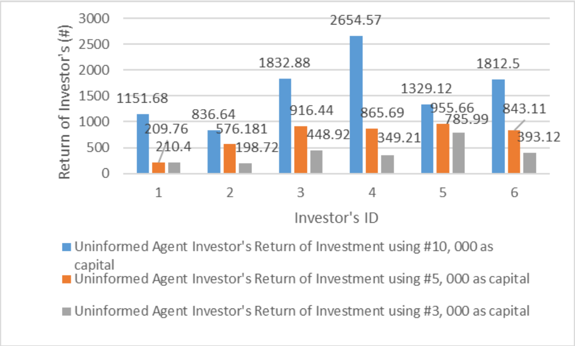


Figure 13: Uninformed Agent Investor's Return of Investment

Six uninformed agent investors were given #10,000, #5,000 and #3,000 each to trade. The return of investment for uninformed investor's agent whose capital were #10,000, #5,000 and #3,000 were shown in figure 13. It was observed that the return of investment for uninformed investor's agent made was far lower than the return of investment made by the informed agent using simple moving average indicator because no indicator was used by the uninformed agent making the agent to be unaware of the save time to buy or to sell. Stock were bought based on the current trading price from the market maker and was sold at a price higher a bit than the price at which the agent got the stock.

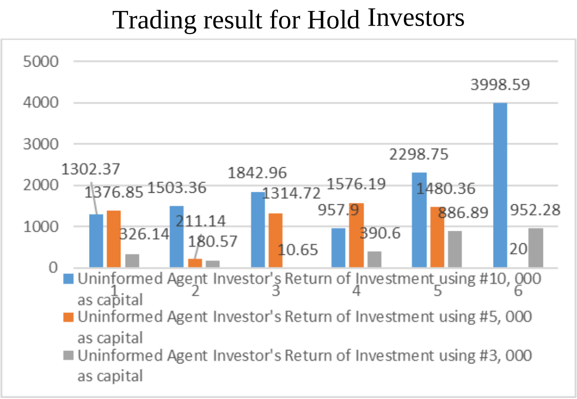


Figure 14: Hold Agent Investor's Return of Investment

Six Hold agent investors were given #10,000, #5,000 and #3,000 each to trade. The return of investment for hold investor's agent whose capital were #10,000, #5,000 and #3,000 were shown in figure 14. It was observed that the return of investment for hold investor's agent made was far lower than the return of investment made by the informed agent using simple moving average indicator. This was so because no indicator was used by the hold investor's agent making the agent to be unaware of the save time to buy or to sell. Stock were bought after a while based on the current trading price from the market maker and was sold after a while at a price higher a bit than the price at which the agent got the stock.

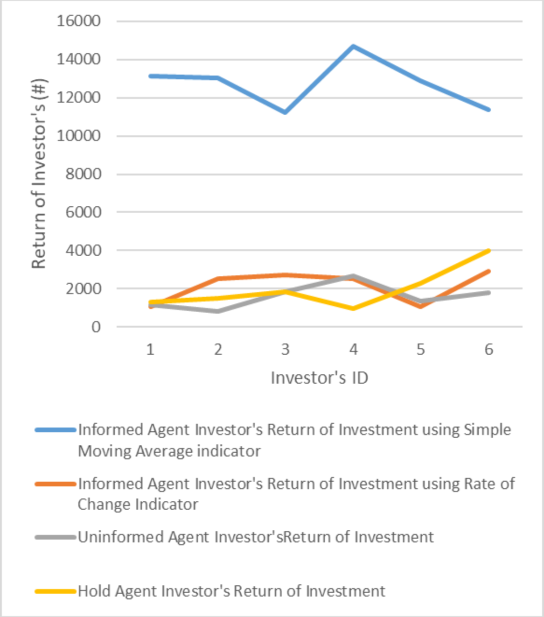


Figure 15: comparing all investor's agent using #10,000 as capital

Comparing the return of investment of all the investor's agent whose capital was #10,000, using informed investor's agent whose chosen indicator was simple moving average has been found to be more reliable to have enough return of investment, than every other investor's agent using such as: informed investor's agent whose chosen indicator was Rate of change indicator, uninformed investor's agent, hold investor's agent as shown in figure 15 because it is shown that there is a wide gap in the return of investment line graph of informed investor's agent using simple moving average and others making informed investor's agent to be at more advantage to the investor's and others to be of less advantage to the investors.

CONCLUSION

In this research, agent is made known as a promising design paradigm for the development of stock trading applications. This work provides a methodical way for implementing a stock trading-based application over the Internet where by the system relies on the utilization of agents as mediators between investors and gives automation support for decision-making tasks and allow users to participate in several marketplaces in a networked e-trading and gives promising solution to the problems of low speed, high latency, and limited computing ability that the current wireless network is facing. This project work has shed more light on the need for agent for stock trading. I therefore recommend that it should be implemented by various stock trading sectors with little or modification in the design to suite their functions and operations accordingly.

a) It is recommended that the specification required for the design of this system should be strictly followed for an effective and functional agent system

b) The project has contributed immensely to knowledge in one way or the other and thus stands as a reference material that can be read and consulted by interested individual departments and units when the topic – Development of an Agent-based Framework for Stock Market Trading is mentioned.

Having covered and addressed many important issues in designing of agent for stock trading system. The followings however are the limitations of this project work;

i. The project work does not provide an avenue for investor's agent to trade on multiple stocks.

ii. Only one market maker agent is allowed to coordinate the trade.

REFERENCES

[1] Aleksander Pivk and Gams Matjaž (2009): Intelligent Agents in E-commerce, Electrotechnical Review, 44-56.

[2] Al-Kasassbeh and Adda Mazd (2007): Analysis of agents in Network fault management, Journal of Network and Computer Applications, 699-711.

[3] Antonio Carzaniga (2000): Designing Distributed Applications with Mobile Code Paradigms, (pp. 22-32), ACM Press.

[4] Aridor Yariv and Lange Danny (2007): Agent Design Patterns, Elements of Agent Application Design, (76-84).

[5] Arnold Marj and Gosling Artsh (1996): The Java Programming Language, ISBN 0201-63455-4.

[6] Baek Samds and Yeom Fraz (2006): A timed agent planning approach for distributed information retrieval in dynamic network environments, Information Sciences (Elsevier Inc.), 3347–3378.

[7] Balon Talg (2006): Preventing Attacks on Agents by MaliciousHosts,14-19

[8] Bieszczad Done, Pagurek Benteng and White Talg (1998): Agents for network management, IEEE Communications Surveys,167-179.

[9] Borselius and Niklas (October 2002): Agent security, Electronics & Communication Engineering Journal, 211-218.

[10] Boutaba and Xiao (2002):"Network Management: State of the Art", Proceedings of the IFIP 17th World Computer Congress, 127-146.

[11] Dasgupta Bansh (1998): A Supplier Driven Electronic Marketplace Using Agents,32-47.

[12] Fabio Bellifemine, Giovanni Caire and Trucco Titesz (2010), JADE PROGRAMMER'S GUIDE, Giovanni Rimassa (University of Parma).

[13] Fung Zanh and Chen Wang. (2005): A multiagent supply chain planning and coordination architecture, International Journal of Advanced Manufacturing Technology, 811–819.

[14] Jain Tehzs and Aparico Satuz (1999): Agents for process coherence in virtual enterprises, Communications of the ACM 42, no. 3, 62-69.

[15] Kachirski Arteh and Guha Rechx (2002): Intrusion Detection Using Agents in Wireless Ad Hoc Networks, Proceedings of the IEEE Workshop on Knowledge Media Networking, 153 -158.

[16] Kalakota Santchij and Whinston Kartz(1996): Frontiers of Electronic Commerce, Addison-Wesley Publishing Company, Inc(87- 96).

[17] Kurt Taraka, Schwehm Lange and Williams Swinburne (1998): Agents, The future of Distributed Computing. IEEE Distributed Communication, vol. 2, no 5, pp34-49.

[18] Lingnau Andrew, Drobnik Tolps and Dömel, P. (Dec 1995). An HTTP-based Infrastructure for Agents. 4th Int. WWW Conference. -

[19] Manvi S, & Venkataram, P. (2004). "Applications of agent technology in communication: a review." Computer Communications, 1493-1508.

[20] Nwana Tarad and Azarmi Cart (1997): Software Agents and Soft Computing, Machine Intelligence, 36-44.

[21] Nigerian Stock Exchange (NSE). (2016): Available online: www.nigerianstockechange.com (accessed on 11 September 2019).

[22] Okonkwo, O.N.; Ogwuru, H.O.; Ajudua, E.I. (2014): Stock Market Performance and Economic Growth in Nigeria: An Empirical Appraisal. Eur. J. Bus. Manag., 33–42.

[23] Oyetunji O.E. and Akintola K.G. (2019). Deploying Mobile Agents for E-Commerce.

[24] Proceedings of the 14th International Conference of the Nigeria Computer Society: Smart Nations,

Digital Economies and Meaningful Lives, July 16-18, 2019, International Conference Centre, Gombe, Nigeria.

[25] Park Tard, Byun Kim and Yeom Argte (2002). The Performance of Checkpointing and Replication Schemes for Fault Tolerant Agent Systems. Proceedings of 21th IEEE Symposium on Reliable Distributed Systems. IEEE Computer Society.

[26] Pullum Larng (2001): Software Fault Tolerance Techniques and Implementation, Proceedings of the IFIP 17th World Computer Congress, 123-135.

[27] Rahul Jha and Iyer Sazerd (2001): Performance Evaluation of Agents for E-commerce Applications. 331-340.

[28] Rebeschini and Mast Mewrg (2004): An Agent Environment for Securing Hosts and Networks, Journal of Network and Computer Applications, 74-88.

[29] Reddy Merfg (2002). Agents Intelligent Assistants on the Internet, Electronics & Communication Engineering Journal, 211-218.

[30] Senn., J. (Dec 2000). The emergence of M-Commerce. 148-150.

[31] Shi, Z., & Zheng, N. (2006). Progress and Challenge of Artificial Intelligence. Journal of Computer Science and Technology, 4-6.

[32] SVitek, J., & Tschudin, C. (1997). Mobile Object Systems: The Programmable Internet. Proceedings of the Distributed Intelligent and Multi Agent Systems Workshop'95, 146-166.

[33] Wang, & Ying-Hong. (2003). Applying Agents to E-business. Tamkang Journal of Science and Engineering, 159-172.

[34] Woolbridge Atharmel and Jennings Nathaniel (1995): Intelligent Agent, Elements of Agent Application Design, 34-44.

[35] Yariv Aridor, Y., & Lange, D. B. (1988). Agent Design Patterns. Elements of Agent Application Design. Proceedings of 21th IEEE Symposium on Reliable Distributed Systems. IEEE Computer Society.