

December 2005

Web-service-agents-based Securities Trading Simulation System

Yingfeng Wang
University of Hong Kong

Huaiqing Wang
University of Hong Kong

Jingjing Deng
University of Hong Kong

Xuemei Zhao
University of Hong Kong

Kayeung Yung
University of Hong Kong

See next page for additional authors

Follow this and additional works at: <http://aisel.aisnet.org/pacis2005>

Recommended Citation

Wang, Yingfeng; Wang, Huaiqing; Deng, Jingjing; Zhao, Xuemei; Yung, Kayeung; and Gao, Shijia, "Web-service-agents-based Securities Trading Simulation System" (2005). *PACIS 2005 Proceedings*. 32.
<http://aisel.aisnet.org/pacis2005/32>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Authors

Yingfeng Wang, Huaiqing Wang, Jingjing Deng, Xuemei Zhao, Kayeung Yung, and Shijia Gao

Web-service-agents-based Securities Trading Simulation System*

Yingfeng Wang
Department of Information System
City University of Hong Kong
Gilbert.wang@student.cityu.edu.hk

Huaiqing Wang[†]
Department of Information System
City University of Hong Kong
iswang@cityu.edu.hk

Jingjing Deng
Department of Information System
City University of Hong Kong
dundask@gmail.com

Xuemei Zhao
Department of Information System
City University of Hong Kong
amethystmeimei@gmail.com

Kayeung Yung
Department of Information System
City University of Hong Kong
tiffany.yung@gmail.cityu.edu.hk

Shijia Gao
Department of Information System
City University of Hong Kong
50095293@student.cityu.edu.hk

Abstract

A global trend towards automated trading systems encourages the appearances of different kinds of decision support systems (DSS) for securities trading. However, the efficiency and effectiveness of the trading decisions and investment strategies suggested by a computer based system are always being oppugned by researchers for abnegating the specialists' professional intermediary roles in traditional floor trading. Without proper validations, evaluations and comparisons, those valuable decision support systems can hardly be adopted trustily. This paper presents a Web-service-agents-based Securities Trading Simulation System (STSS) as a platform to validate, evaluate and compare the existing securities trading decision support systems by using the real world securities trading data from Chicago Stock Exchange. Agent technology is applied to deal with the complex, dynamic, and distributed securities trading processes. Web-services techniques are proposed for more interoperability and scalability in network-based business environment. By integrating agent technology with Web-services to make use of the advantages of both, this approach provides a more intelligent, flexible, and comprehensive platform to simulate the securities trading process. Performance of different decision support systems for securities trading can be validated, evaluated and compared either separately or simultaneously on a real time basis.

Keywords: Intelligent agents, Web-services, Securities Trading Simulation, Decision Support System.

1. Introduction

* This paper is supported by a Strategic Research Grant (No. 7001805) from the City University of Hong Kong

[†] Corresponding author. Tel.: +852-2788-8491; fax: +852-2788-8535

Modern securities markets have become heavily dependent on technology. The New York Stock Exchange (NYSE) has invested \$1 billion in information technology since 1978 to increase capacity and support trading¹. Exchanges, independent of their size, are adopting technology to automate trading (Weber et al. 2002). Under this circumstance, numerous different kinds of DSSs for securities trading in almost every aspect, including risk management, technical analysis, portfolio management, etc (Chou et al. 1996; Luo et al. 2002; Wang et al. 1997; Wang et al. 2004), were developed by researchers. Based on long run simulation, some systems were claimed to outperform their counterparties like NYSE or others systems. However few researches have been conducted on evaluating or comparing these systems under real world trading market circumstance. Little has been known about what kind of system performs better under what kind of market circumstance. Therefore no effective suggestions provided to the existing DSSs, ongoing DSSs constructions and end-users on better fitting criteria for DSSs. However, by simulating the real world trading processes, these targets can be achieved. After results are adequately anatomized, above issues may be proper addressed.

Real world securities trading are a complex and dynamic process, in which different parties are competing for profits from tremendous interactions. To simulate such kind of complex human behaviors, the system has a high degree of both autonomy and cooperative capabilities. Agents-based systems are considered more suitable under such situations (Dugdale et al. 1996; Jennings et al. 1998; Wooldridge et al. 1995). Therefore, an agents-based approach is proposed in this research. Given specific knowledge and capabilities, intelligent agents are capable of dealing with complex problems and vast amounts of information collaboratively in dynamic and unpredictable environments (Wang, et al. 2004). Moreover, web services are adopted as promising technology to both support open and distributed agent components for cooperatively working together, and ease the access of those DSSs.

Two months future trading data of 2004, including trade price and ask-bid queues, from Chicago Stock Exchange (CHX) is acquired as the simulation environment.

The rest of the paper is organized as follows. Next Section briefly reviews the relevant literature on automated securities trading, securities trading simulation systems, intelligent agent theory, Web-services and integration of intelligent agent-based systems and Web-services. In Section 3, the architecture design of the proposed Securities Trading Simulation System (STSS), which includes the intelligent agent hierarchy, Web-service-agents-based system architecture, and multi-agent interaction, has been presented. The development of the STSS prototype system is described in Section 4. In order to evaluate the prototype system in Section 5, a use case is created to illustrate the effectiveness of the system. Finally, Section 6 addresses our contribution as well as the future work.

2. Background

2.1 Automated Securities Trading

Automated trading, from the early time, has been claimed to be able to enhance the efficiency of the market's price discovery function and liquidity, transparency contributes to the efficient allocation of scarce capital among competing demands for that capital (Becker et al. 1993). However, as a decision maker, it has also been criticized on that excluding human efforts will

¹ Edward A. Kwalwasser, Executive Vice President, Regulatory group, New York Stock Exchange. Testimony before the Subcommittee on Telecommunications and Finance, May 26, 1993.

not only reduce the effectiveness of the decision making (Benveniste et al. 1992) but also affect efficiency, in terms of transaction cost (Venkataraman 2001), which is usually regarded as advantages of automated trading (Domowitz & Steil 1999) for decades. Actually no conclusions are addressed to be widely agreed upon this debate. However, what for sure is that without proper decision support, using automated trading runs the risk of making wrong decisions and losing the principal benefit of the mechanical approach (Wong et al. 1994).

2.2 Securities Trading Simulation Systems

From the literature of Securities Trading Simulation, many research outputs have been achieved (Lin et al. 2004; Næs et al. 2003). However, focusing either on the simulation strategies and functionalities or on analyzing the results of simulation cannot give comprehensive understanding on the performance of different DSSs under different situations. Therefore, it cannot provide effective validations, evaluations and comparisons to DSSs.

Only a few works provided the completed implementations on STSSs, and how STSSs work. Among them the best example with growing platform and system components is perhaps the popular and successful Trading Agent Competition (TAC) (Kearns & Ortiz 2003). However because of different objective of design, the TAC systems (Stone et al., 2001 Wellman et al., 2001; Wellman et al., 2003) are focusing on providing a platform for trading competitions among human beings and study the human behaviors through such competitions. Thus, historical data support and interface with other systems are unnecessary in TAC systems, which undermined their abilities to perform the validation, evaluation, and comparison tasks. Therefore, providing suggestions to existing DSSs, ongoing DSSs construction and providing end-users with criteria on how to choose a DSS as their investment decision supporting partner remain blank.

2.3 Agents and Multi-Agent Systems

An agent is a computer system situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives (Jennings et al. 1998; Wang 1997). A multi-agent system (MAS) consists of a group of agents, interacting with one another to collectively achieve their goals (Gao et al. 2005).

Considering the flexible, distributed, and intelligent solutions provided by agents-based system, to business applications, a considerable growth of interest amount researchers has been recognized, in terms of numerous design and development multi-agents-based systems have been proposed. (Dugdale et al. 1996; Murthy et al. 1999; Wang et al. 2002; Wang et al. 1997).

By using a society of agents, where an agent is considered a trading process responsible for a subset of possible scenarios of the way system functions on behalf of human being, automated trading can enjoy easy realization; perform limitless, error-free routine calculations (Wang et al. 2002) and reduction on operational overhead dealing with various related functionalities such as withdrawing a order or modifying the unexecuted order.

2.4 Web-services

That numbers of network-based business services are conducted using web services shows a remarkable utilities provided by web services. Web services promise to take virtually any software function, legacy or newly developed, and wrap it in an abstract interface that allows it to be performed on behalf of any entity via simple XML messages delivered over the Web (Silver 2002).

2.5 Integration of Intelligent Agent-based Systems and Web-services

Typical agent architectures have many same features as Web-services, and enhance Web-services in several ways: (Huhns, 2002)

- Web-services, unlike agents, are not designed to use and reconcile ontologies;
- A Web-service knows only about itself, while agents often have awareness of other agents in the same environment including their behaviors, capabilities, though interactions among the agents occur;
- A Web-service, as currently defined and used, is not autonomous. However, as defined before, autonomy is a basic characteristic of agents;
- Agents are inherently communicative, whereas Web-services are passive until invoked;
- Agents are cooperative, and by forming teams and coalitions can provide high-level and more comprehensive service, while current standards for Web-services do not provide for composing functionalities.

3. Architecture Design of STSS

The design of a novel agent-based securities trading simulation system is described in this section.

3.1 Deployment of Intelligent Agents

Before applying multi-agent technology into STSS, we need to decompose the process of securities trading into several autonomous stages, where each agent is delegated a particular task to exhibit its goal-oriented and reactive behaviour, and cooperate with others to pursue their goals.

The Securities Trading Simulation System (STSS), as a DSS based automated trading system, is to provide a running platform for participants to place buy and sell orders, based on an environment of historical data from Chicago Stock Exchange (CHX.).

Therefore, the whole process can be divided into two parts. One is mainly focusing on the routine orders and transactions invoked by the environment; the other is mainly focusing on the interaction with simulation participants. However, when participants join in the simulation, influences on the environment made by their unpredictable behaviour are foreseeable. Here come the challenges for the multi-agents system.

Based on above rationality, several basic task-oriented agents are defined, including Data Acquisition Agent (mainly focusing on dealing with the incoming environment data), Client Interaction Agent (playing the role as a access port of participants, perceived knowledge on other agents' capabilities is required for rapid and accurate interactions with outsiders), Performance Monitoring Agent (Focus on monitoring the performance of all participants, and responsive to inquires on these information to all the other parties), Order Handling Agent (handling vast inactive limit orders, manage the ask and bid queues, by sensing the possible price movement or receiving request from Simulation Management Agent, supply top asks and bids with indicators on the order owner based on certain rules), Simulation Management Agent (performing a transaction confirmation role, with less but crucial order book information like most possible executable orders and market order, as well as a role of information supplier, not only to Performance Monitoring Agent but to all the other parties for reporting the commitment of transactions).

3.2 System Architecture

Based on the agents deployment above, complex and dynamic securities trading processes are delegated to a society of autonomous agents, which behave on behalf of different specialists involved in those processes with perceiving the corresponding environment and specific knowledge. These agents are wrapped as Web-services, communicating and interacting with each other in an open environment.

Figure 2 presents the architecture of the Web-service-agents based STSS.

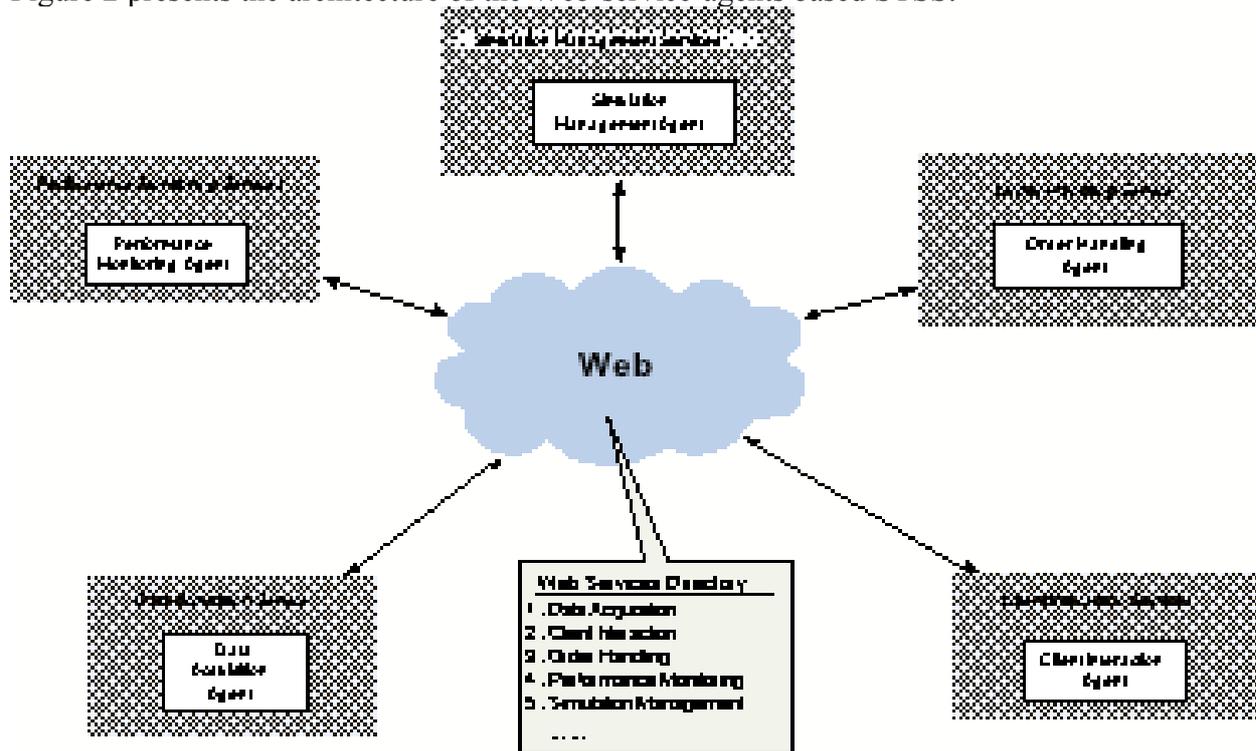


Figure 2. STSS Architecture

In this architecture, all these Web-services agents work autonomously and collaboratively via the Internet. Each agent will mainly focus on its particular task, such as data acquisition, client interaction, order handling simulation management, without inventions from outside. There might be one or more agents involved in a particular service. By drawing on other agents' knowledge and capabilities, agents can overcome their inherent bounds of intelligence and work collaboratively to pursue their goals.

4. System Implementation

The STSS's web-service agents provide securities trading services on the internet by using the real world historical data.

Java IO is used for the data acquisition program. Due to the fact that the data flow from the text file is one-pass at the initial stage, we adopted sequential reading/writing with buffer. This program runs as a thread to provide potential multi-threading feature in future development.

The most popular language for agent communication is Knowledge Query and Manipulation Language (KQML). Recent research has focused on the use of XML (Extensible Markup Language) in agent communication (Glushko et al, 1999). The Web-service agents have been developed by using Java Web-services Development Package (JWSDP) (Java.sun.com, 2004), which brings together a set of Java APIs for XML-based Java applications by supporting key XML standards, such as SOAP (Simple Object Access Protocol), WSDL (Web-services Description Language), and UDDI (Universal Description, Discovery and Integration). These APIs and their reference implements are bundled together with a set of runtime tools to form a JWSDP. Web services use the popular Internet standard technology – SOAP – to increase compatibility of the system. SOAP is the most common network communication protocol between software services.

Moreover by using SOAP and other add-on libraries such as JAVATM MAIL, Junit and activation JARS for data transmission, the program can transfer the processed data to any Web Services clients as a text file attachment, or receive attachments, and this approach makes sure the data manipulation at the client side is simple and efficient.

The major advantages of using SOAP with Attachments (SwA) are that sending and receiving a SOAP message using standard SOAP takes much longer than that carries the same data using SwA, and the size of the SOAP message built by standard SOAP is several times larger than when using SwA. This shows the use of SwA to deal with complex datatypes and large amounts of data can improve performance by reducing XML parsing.

KQML is wrapped by SOAP. SOAP messages are represented by using XML, and can be sent over a transport layer, such as HTTP and SMTP. In the prototype system, the communication amongst agents is done by Java API for XML Messaging (JAXM). Such JAXM messages follow SOAP standards, which prescribe the format for messages and specify the things that are required, optional, or not allowed.

The Web-service agents are able to detect and resolve the conflicts by JESS (Java Expert system Shell and Scripting language) rules. JESS is a rule engine and scripting environment written entirely in Java language (JESS, 2004). In the prototype system, each agent contains a JESS rule set for reasoning. The reasoning results are asserted JESS facts. An agent can send such facts to other agents, by wrapping them to XML and SOAP attachments. After the Web-service agents have been set up, they should be published on the Web. The WSDL specification is used to describe and publish Web-service agents a standard way. Finally, the UDDI is used to create and implement a directory of the Web-service agents

5. System Evaluation by a Use Case

In order to demonstrate the effectiveness of proposed approach and illustrate how the Web-service agents work together to reach the simulation goals, the simulation operation process of their collaboration is illustrated in the following case. Performance of different decision support system for securities trading can be validated, evaluated and compared either separately or simultaneously on a real time basis.

Several rules and assumptions before starting

- Simulation on security trading must be running before any DSSs can participate.
- Although the historical data available now is text file-based, as simulation we would take it as real-time incoming data because with same data structure the data acquisitions process is the same.
- Since only having 2-month data, simulation will be conducted continuously.
- Performance measurements will be based on profitability, in term of the total value including on hand cash and security value.
- Transaction cost will be charged like CHX.
- Performance is traced through the whole simulation, which means DSSs can request to viewing their current performances comparing with others any time during the simulation.
- Short selling is allowed because of in the real world short selling is also a good instrument to seek the opportunities, however, the interest will also be charged.
- No initial money provided, since short selling is allowed and DSSs should be able to make profit though investments instead of requiring initial funds.

Figure 3 presents the collaboration and iteration within the STSS web-services agents.

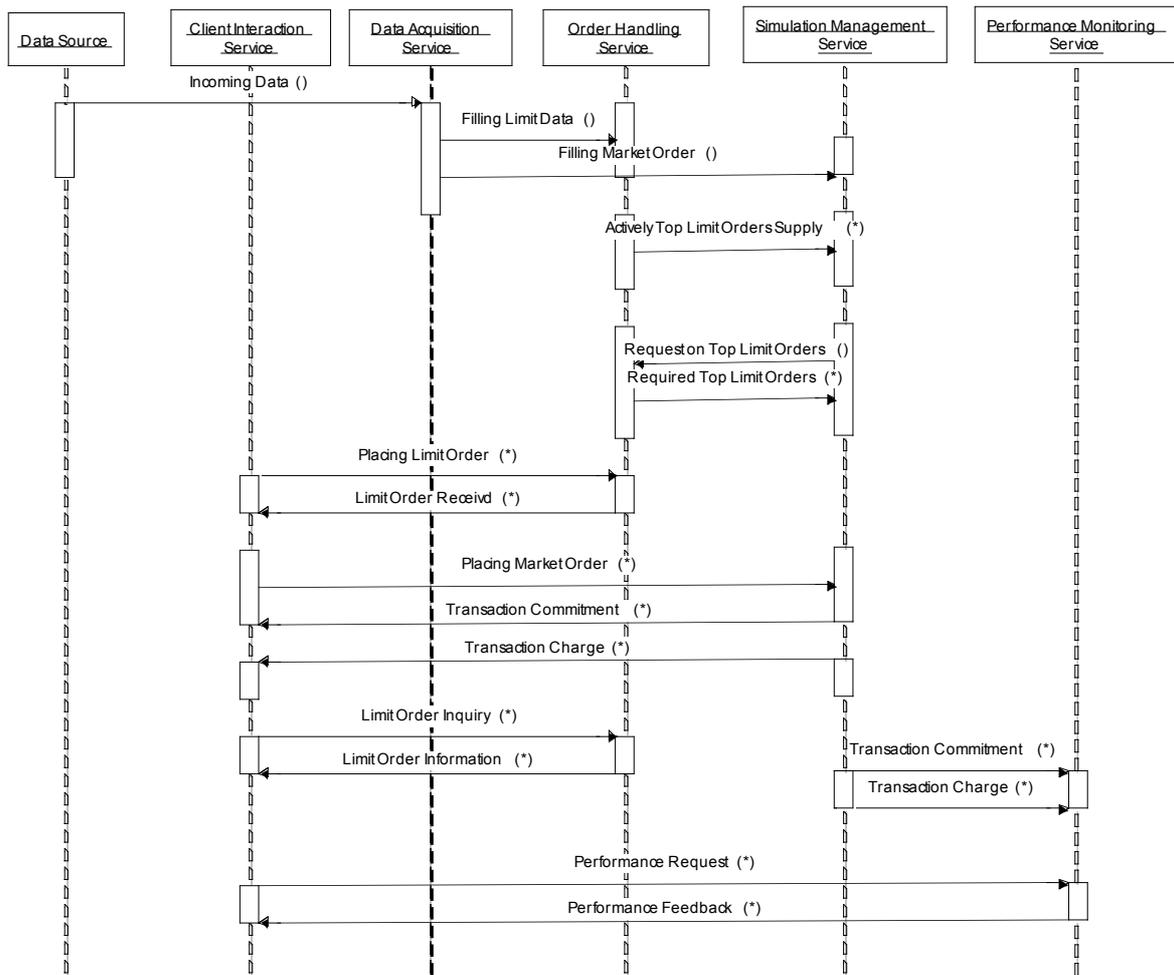


Figure 3. Collaboration and Interaction with STSS Web-service Agents

Assume that only one DSSs participants the simulation to validate and evaluate its profitability.

1. Data Acquisition Service begins to do continuous routines, receiving environment data from data source and filling the order to different services based perceived knowledge.

Simulation started...

2. Participant DSS 1 (D1) requests for Client Interaction Service through internet and begins to join in the simulation after registered and get a Client Identification (CI). (Since number of participants is unpredictable, it is not reasonable to provide each incoming DSS an agent. Therefore participants are unique to STSS by using their CIs).
3. Client Interaction Service places different orders on behalf of D1 using its CI. Different orders go to different web-services for further operation.
 - Limit orders go to Order Handling Service different from the order filled by Data Acquisition Service, the star in the brackets indicate D1's CI. After the order is seated properly in the order book, feedback on receiving the order also based on CI comes back. And now Order Handling Service keeps a limit order with CI indicator from D1.

- Market orders go to Simulation Management Service and executed immediately, all parties are informed of the transaction details and charges, but only Performance Monitoring Service keep the data.
4. An example of the collaborative between Simulation Management Service and Order Handling Service is explained as in the Figure 3. When receiving the transaction detailed information from Simulation Management Service, Order Handling Service feels that trading price is quite close to the top ask in its hand, then proactive agent-based communication between web-service occurs, Order Handling Service actively provides Simulation Management Service with several highest ask limit orders with the CI indicators. Another example shows here is when limit order in Simulation Management Service is lower to some extend based on some rules, it will automatically request highest limit order from Order Handling Service.
 5. Client Interaction Service places request on the unexecuted order information, require changing the order or even withdrawing the order on behalf of D1. The informational and operational functions are performed reactively by different web-service.

For Multi-participants competition situation, everything addressed above is the same except that the CI indicators carried on are different, during the simulation.

6. Conclusion and Future Work

In this paper, a novel architecture of Web-service-agents-based securities trading simulation system has been presented. Multiple agents with different service are wrapped as Web-services and capable of communicating with each other via standard Web-service protocols. The main contribution of this research can be summarized below:

- From the technology perspective, a novel and open architecture for STSS has been designed. By integrating agent technology with Web-services to make use of the advantages from both, this approach leads to more intelligence, flexibility and collaboration in securities trading simulation. Different Web-services can cooperate together to provide comprehensive trading facilities. Each Web-service can be separated to function as a stand-alone system, select and call other agents' service freely through Internet, if required. In addition, it is easy to add more business functionalities into our STSS by adding more Web-service agents. And all Web-service agents are reusable by other business applications.
- From business perspective, currently existing securities trading simulation systems mainly focusing on either providing better decision support or human competition environments for behavior study. Real world security trading situations are full of complexity and uncertainty. Compute-based decision support systems enjoy a lot of advantage when competing with humans in seeking and grasping the investment opportunities in the real world situation. However, such systems will not adequately behave under all circumstance due to the tremendous diversity of human society. Without proper evaluations on these DSSs, effective suggestions can hardly be provided to the existing DSSs, ongoing DSSs constructions and end-users on better fitting criteria for DSSs. By applying multi-agent technology with combination of the Web-services, the STSS presented in the paper can perform as a well established securities trading platform. With this platform DSSs for security trading could be validated, evaluated and compared by competing with competitors within real world data supported environments. Therefore, the STSS presented here provides an effective stage for the problems issued to be properly addressed.

Future works could be focused on an experiment based evaluation to determine the effectiveness of our STSS, as well as its real world application.

References

- Becker, B., Lopez, E., Berber-Doumar, V., Cohn, R., and Adkins, A. S. "Automated Securities Trading," *Journal of Financial Services Research* (6:4), 1993, pp. 327-342.
- Benveniste, L. M., Marcus A. J., and Wilhelm, W. J. "What's special about the specialist," *Journal of Financial Economics* (32:1), 1992, pp. 61-86.
- Chou, T. SC, Yang, CC, Chan, CH, and Lai, F. "A Rule-Based Neural Stock Trading Decision Support System," *Proceedings of the IEEE/IAFE 1996 Conference on Computational Intelligence for Financial Engineering*, New York, 1996, pp. 148--154..
- Domowitz, I. and Steil, B. "Automation, Trading Costs, and the Structure of the Trading Services Industry," *Brookings-Wharton Papers on Financial Services*, 1999.
- Dugdale, J. and Keynes, M. "Cooperative problem-solver for investment management," *International Journal of Information Management* (16:2), 1996, pp.133-147.
- Gao, S., Wang, H., Wang, Y., Shen, W., and Yeung, S. "Web-service-agents-based Family Wealth Management System," *Expert Systems with Applications* (In press)
- Glushko, R. J., Tenenbaum, J. M., & Meltzer, B. "An XML framework for agent-based e-commerce," *Communications of the ACM* (42), pp. 106-114.
- Huhns, M. N. "Agents as Web services," *IEEE Internet Computing* (6:4), 2002, pp. 93-95.
- Java.sun.com, Java Web Services Developer Pack 1.4, 2004,
<http://java.sun.com/webservices/jwsdp/index.jsp>
- Jennings, N. R. and Wooldridge, M. *Applications of Intelligent Agents*, In: Agent Technology: Foundations, Applications, and Markets, 1998, pp. 3-28.
- JESS – the Rule Engine for the Java™ Platform (2004). <http://herzberg.ca.sandia.gov/jess/>.
- Kearns, M. and Ortiz, L. "The Penn-Lehman Automated Trading Project," *IEEE Intelligent systems* (18:6), November/December 2003, pp. 22-31
- Lin, CW, Shih KH, and Huang, S. Y. "How is Market Efficiency Disappeared? Comparing the Opening Position and Closing Position Simulation Results," *Journal of American Academy of Business, Cambridge* (4:1/2), Hollywood, March 2004, pp. 314.
- Luo, Y., Liu, K., and Davis, D.N. "A Multi-Agent Decision Support System for Stock Trading" *IEEE Network* (16:1), January/February 2002, pp. 20 - 27
- Næs, R. and Skjeltorp, J. A. "equity trading by institutional investors: Evidence on order submission strategies," *Journal of Banking & Finance* (27:9), September 2003, pp. 1779-1817
- Schwartz R. A. and Weber B. W. "Next-Generation Securities Market Systems: An experimental investigation of quote driven and order driven trading" *Journal of Management Information System* (14:2), 1997, pp. 57-79.
- Silver, B. "Choreography: The next step for Web services," *Transform Magazine* (11:8), ProQuest Computing, Aug 2002, pp. 19.
- Stone, P., Littman, M. L., Singh, S., and Kearns M. "ATTac-2000: An Adaptive Autonomous Bidding Agent," *Journal of Artificial Intelligence Research* (15), September 2001, pp.189-206.
- Venkataraman, K. "Automated Versus Floor Trading: An Analysis of Execution Costs on the Paric and New York Exchange," *The Journal of Finance* (56:4), August 2001, pp. 1445-1485
- Wang, H., Mylopoulos, J., and Liao, S. "Intelligent Agents and Financial Risk Monitoring Systems," *Communications of the ACM* (45:3), pp. 83-88.
- Wang, H. "Intelligent agent assisted decision support systems: integration of knowledge discovery, knowledge analysis, and group decision support," *Expert Systems with Applications* 12(3), 1997, pp. 323-335.

- Wang, H. and Wang, C. "Intelligent Agents in the Nuclear Industry," *IEEE Computer* (30:11), 1997, pp. 28-34.
- Wang, JH, Chen, SM, and Leu, JY "Stock trading decision support system using a rule selector based on sliding window" In *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 559-564,
- Wang, M., Wang, H., Xu, D., and Vogel, D. "A web-service agent-based decision support system for securities exception management," *Expert Systems with Applications* (27:3) Oct. 2004, pp. 439-450.
- Weber, B. W. and Levecq, H. "Electronic Trading Systems: Strategic Implications of Market Design Choices," *Journal of Organizational computing and Electronic Commerce* (12:1), 2002, pp. 85-103.
- Wellman, M. P., Wurman, P. R., O'Malley, K., Bangera, R. Lin, S., Reeves, D., and Walsh, W. E. "Designing the Market Game for a Trading Agent Competition," *IEEE Internet Computing* (5:2), March/April. 2001, pp. 43-51.
- Wellman, M. P., Greenwald, A., Stone, P., and Wurman, P. R. "The 2001 Trading Agent Competition," *Electronic Markets* (13:1), 2003, pp. 4-12.
- Wooldridge, M. and Jennings, N. R. "Intelligent agents: theory and practice," *The Knowledge Engineering Review* (10:2), 1995, pp.115-152.
- Wong, R.K. and Ng, P.N. "A hybrid approach for automated trading systems," In *Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems*, 29