Australian Journal of Basic and Applied Sciences, 4(3): 442-449, 2010
ISSN 1991-8178

A Research Survey of Software Agents and Implementation Issues in Vulnerability Assessment and Social Profiling Models

Ghulam Ali, Noor Ahmed Shaikh and Abdul Wahid Shaikh

Department of Computer Science, Shah Abdul Latif University, Khairpur

Abstract: This paper discusses theoretical and practical aspects of software agents. A research survey regarding industrial applications, multi-agent systems, platforms, standardization and classifications has been presented. Various models of Network security, specifically vulnerability assessment model and Social profiling involving software agents have been studied and compared in the research. The designing and implementation issues of agent frameworks are major concern. Finally the appropriate performance evaluation methodology to conduct practical on software agents has been proposed on different datasets.

Key words: Software Agents, Vulnerability Assessment, Insider Threat, Social profiling, Agent Framework

1. Theoretical Aspect of Software Agents:

Researchers, so far, have not been able to agree upon a common definition for software agents. However, different researchers have defined agent according to their own point of view. Some researchers have described software agent as "an umbrella term for a heterogeneous body of research and development" (Hyacinth S. Nwana, 1996). Software agents are not merely confined to computer science but also involve diverse fields such as sociology, psychology, etc (Tosic et al, 2004). Domain-specific definitions have also given by researchers. According to the definition of Russell and Norvig (Stuart Russell and Peter Norvig, 2002) agent performs two tasks: It senses its surrounding environment through sensors and performs actions in it with its effectors. According to another definition "Autonomous agents are computational systems that inhabit some complex dynamic environment; sense and act autonomously in this environment and by doing so realize set of goals or task for which they are designed" (Maes and Pattie, 2005). Some researchers view agents as special software that involve in communications, bargaining, coordination, and perform so many other actions autonomously same as being done in real life (Michael H. and Coen, 2004; Brustoloni and Jose C., 2003).

1.1 Characteristics and Classifications:

Agent possesses few or all properties as suggested by Etzioni (Etzioni, O., and Weld, D. S., 2005) depending on the requirement of the problem. The major focused properties are Autonomy, Reactivity, Collaborative behavior, Social Behavior, Adaptability and Mobility. The research on software agents is taking place in three different dimensions. The partial work has been done in almost all three domains of software agents, shown in figure 1. In Agent Oriented Programming (AOP), FIPA-compliant agent framework has been developed where autonomous agents are built at application layer. In Artificial Intelligence, autonomy, reactivity, and social ability have been investigated through agent based context aware environment where social communities are developed on the basis of common interests.

Graesser classifies agents on the basis of the properties they possess (Franklin, S., and Graesser, A., 1996). The major properties may include reactiveness, autonomy, goal-driven, temporal continuity, communicative, learning, mobility, flexibility, etc. Every agent possesses first four properties. Russell & Norvig classify agents into four categories: Simple Reflex Agents, Reflex Agent with state, Goal-based Agents and Utility-based Agents. According to the Brustoloni's classification, agents have been categorized as Regulation Agent, Planning Agent and Adaptive Agent. J. Alfredo Sánchez categorizes agents on the basis of their use in different domains, i.e. Programmer Agents, Network Agents and User Agents (J. Alfredo Sánchez, 1997). After study of various characteristics of software agents we propose following classification, shown in figure 2.

Corresponding Author: Ghulam Ali, Department of Computer Science, Shah Abdul Latif University, Khairpur.
E-mail: ghulam.ali@salu.edu.pk

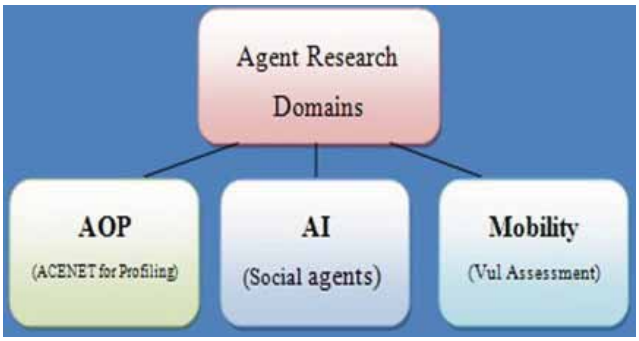

Fig. 1: Research domains in software agents

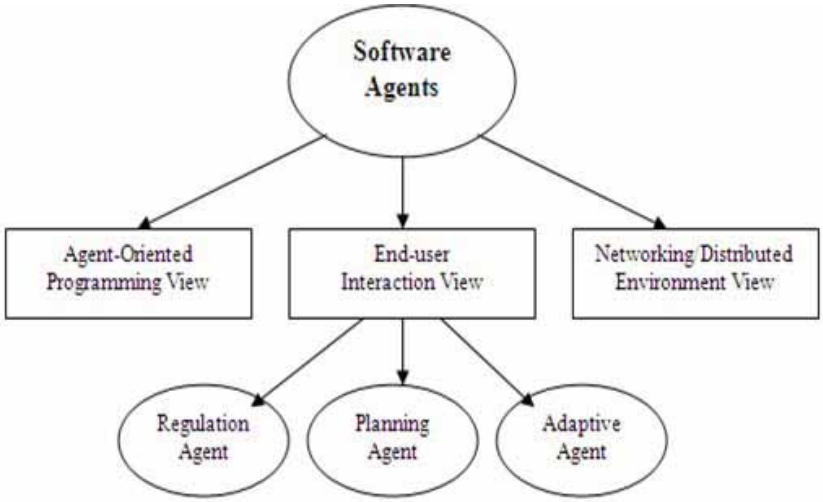

Fig. 2: Proposed Classification of Software Agent Research Domains

Software developers use agent oriented programming view and network view as abstractions to cope up with the designing complications. The major advantage of agent-oriented programming is to create a hierarchy so that the complications may be reduced. ACLs, Interaction protocols and Agent Markup Languages have major contribution in agent programming abstractions. In distributed or network view, agents are autonomously migrating entities that act on behalf of network nodes. For the execution of the agents and their security on the network, management tools or programming environments have been developed. Aglets, TACOMA and Telescript (Johansen, D., 2002; Niwano and Okamoto, 1997; Dijon, France, 2006) are the projects developed for the management of the mobile agents. In end-user interaction view, agents have been proposed as an abstraction for end-users to interact with computer systems. End-user gets full advantage of autonomous and decision-making capabilities of computer programs. This agent class falls into the category of Artificial Intelligence. This agent can further be classified as Regulation, Planning and Learning agent. Regulation agent watches the input and reacts accordingly because it is aware of its actions. This type of agent does not plan and even it has no capability to learn its environment. Planning agent is different from regular agent in the way that planning is included in the body of this agent. The third type of AI agent is Adaptive agent that has the capability of planning as well as learning.

1.2 Applications:

The research of software agents is being carried out in academic institutes as well as in reputable commercial organizations. Small and large agent-based industrial applications have been designed; tested and deployed, are running according to the needs of the market (Gilbert et al, 1995). The noticeable applications are discussed below.

Network Administration: Agents can be used to boost network's management to help them in taking autonomous actions at higher level.

Roaming: Due to portability users are becoming more invasive and want to use more and more services without considering limitations of the technology such as bandwidth. Software agents moving over network considers users requirements and try to accomplish them, as it cannot be simply achieved without agents.

E-mails: Agents handle emails and other messages of the users as user wishes to operate them. The mails are autonomously filtered by agents as per wish of the users.

Workflow and Administrative Management: Administrative management includes both workflow management and computer/telephony integration. Agents can be used to ascertain, and then automate user wishes or business processes.

E-commerce: Now-a-days agents are being exploited in handling ecommerce applications. Agents are negotiating on behalf of buyers or sellers.

Jennings (N. R. Jennings and M. Wooldridge, 1998) has categorized agent applications on the basis of technology such as industrial, commercial, medical and entertainment.

1.3 Multi-agent Systems:

When single software agent is unable to address a problem because of its inadequate competence, then several software agents form a distributed loosely coupled network and work together to solve the problem, such management is called Multiagent System (MAS). MAS are used in such applications and problem solving that are distributed in nature. In Multiagent systems task is decomposed into several subtasks; subtasks are distributed among various agents, and agents interact with each other. Because different agents are of the different capabilities, therefore there is need of building a system as agents may cooperate, coordinate, negotiate to achieve the task (Zhong Zhang, 2004). The rationalization of the Multiagent system is (a) each agent has partial knowledge for addressing problem; (b) missing of total control on the system; (c) data is decentralized; and (d) communication is asynchronous (Katia P. Sycara, 1998). There is need of an architecture that manage all activities of the agents, such as life cycle management of the agents, coordination mechanisms, the services that agents provide so that agents may know the capabilities of each other, communication languages so that agents may communicate each others with a common language (J. Lind, 2001).

1.4 Agent Platforms:

An agent platform is architecture to provide environment to Multiagent systems for management, communication, negotiation, coordination among agents. There are two types of the agent platforms; one is for implementing and deployment of multi-agent systems which includes JADE, FIPA-OS, Zeus, etc (AgentBuilder, 2000). The other is for the execution and roaming of the agents and their security on the network such as TACOMA, Telescript, Aglets, etc (Danny B. Lange and Mitsuru Oshima, 1997; Johansen, D., 2002; Niwano and Okamoto, 1997). Software agents can perform many activities and even send messages to ensure communication through agent platforms. Amongst agents communication along with ACL an appropriate ontology is also defined for Multiagent systems. Some inter-agent coordination strategy is also included in the agent platform. As agent platforms increase, the need to standardize them to resolve interoperability issues has also been increased.

1.5 Agent Standardization (A Comparative Study):

With the increasing number of Agent development frameworks for heterogeneous systems, the need for the standardization arose. This is because of the incompatibility issues faced when agents of one framework tried to interact with agents of some other framework (Frank Manola, 1998). FIPA and OMG started to work on standardization during 1990s. However, because of these two organizations working independently with no collaboration among them, the end result was the development of two parallel and competing standards, FIPA and OMG-MASIF, each providing entirely disjoint features (Christos Georgousopoulos and Omer F. Rana, 2002).

FIPA specifications are based on remote communication services and do not say much about mobility while MASIF enables agents to move from one host to another and does not address inter-agent communication. FIPA-compliant platforms can express cooperation in better way as compared to MASIF-compliant platforms. MASIF-compliant platforms are appropriate in dynamic swapping, replacement, updating of the software components. However, there are some features, i.e. Security, Distributed Events, Multicasting and Continuation that are not addressed at all by any of these two standards. There is need of the combination of these two standards that may address all features that is necessary for a true Agent System (Menelaos, et al, 1999; Kaffille S. and Wirtz G., 2003). The features comparison of both standards is shown in table 1.

Table 1: Feature Comparison of FIPA and MASIF Standards

Feature	FIPA	MASIF
Proposed By	FIPA	OMG
Communication Language	ACL	None
Interaction Protocols	Basic	None
Granularity of Communication	Message	Mobile Agents
Multicasting	No	No
Distributed Events	No	No
Migration	No	Yes
Tracking	No	Yes
Continuation	No	No
Syntactic Interoperability	FIPA Compliant Systems	Same Agent System Type
Semantic Interoperability	Yes	No
Security Features	No explicit mechanism specified	Based on Corba-IDL, needs more efforts on security
Yellow Pages	Directory Service	MAFFinder Interface
Agent Management	FIPA AMS	MAFAgentSystem
Products	JADE, ADK, April, FIPA-OS, KEAP, ZEUS	Aglets, Grasshopper

2. Vulnerability Assessment and Insider Threat:

Security is generally defined as "the condition of being protected against danger or loss" (M. W. Tobias, 2000; Charles P. and Shari Lawrence, 2003). The circumstances or conditions that have potential to cause loss or harm the computing systems are called threat. Vulnerability is the weakness in the system and its exploitation is known as attack. As computer system is made up of three valuable components: hardware, software and data. There is always threat for these three components whose vulnerably can be exploited by the attacker. Confidentiality, integrity and availability of the computing system components are the systems that are tried to be ensured. Confidentiality tries to ensure that computing resources may be used by the persons who have right to use. It could be used as synonymous for privacy. Integrity ensures that computing resources can be modified only by the persons who are authorized to do. Availability guarantees that only authorized persons have right to access computing resources at any time. An appropriate secure system will always ensure all said tasks. There is need to build a secure system that ensure balancing when any kind of conflict arise. For example, more confidentiality can affect availability. Four types of the attacks on the computing systems have been identified, i.e. interception, interruption, modification and fabrication. When confidentiality is compromised this kind of attack is called Interception that alarms that some person, who has not right to access the resources, has gained the access. In an interruption, the computing resources become unavailable and in result availability compromised. The integrity will be compromised with attack of modification where an unauthorized person makes changes in the data. Fourth problem that might be encountered is the fabrication threat where attacker poses as someone (Wang Zeng-quan et al, 2006).

The area of vulnerability assessment is not new, but the use of mobile agents, is still an evolving area. Many researchers are working in the area either commercially or academically to enhance the security of the systems. Few projects or models are presented that are related to our work. Their limitations and difference of these models from our work are also pointed out. The related work presented here is related to intrusion detection and insider threat. Our major focus is on insider threat because the attacks made in this way are generally initiated by trusted and authorized users that are not immediately traceable. Filtering the traffic at the time of access is not totally successful (Pikoulas, J. and Buchanan, 2002). To increase defense capability the use of other technologies can also be considered. Here, mobile agents are presented to provide computational security by constantly moving around network. The proposed system watches user activities in instantaneous time and take suitable actions when required. It uses a short-term forecasting to predict the user

behavior and advises the administrator accordingly, before the actual actions take place. The proposed model uses the collected user data and produces a profile to foresee the future user action on the basis of the information that system has acquired. The related work has been done by few researchers but their limitation is disappearance of autonomy (Jamal Bentahar et al, 2006). It would give better results if agents are introduced. As discussed earlier, most of the security related projects and solutions focus on outside threats. Very little research has been so far carried out on inside threat. The major limitation, that has been identified, is that most of the solutions focus on technology rather than user behavior. But research has proved the significance of user behavior along with technology. Both domains are equally important for efficient security.

It seems that there is complete lack of research on agent framework that may work autonomously and follow some standards to achieve the overall goal. We propose autonomy and pro-activeness in the model that is least focused in the related work.

3. Social Profiling Model:

Social profiling is not purely domain of computer science rather it has roots in various areas such as psychology, social sciences, cognitive sciences and many other areas. The intention in this research is to focus on computer science therefore papers, related that particular domain, have been referred. Following are the projects and research work that is related to the proposed model.

For agent-human communication a framework, POSTAGE, was developed to involve software agents for communications with human (Boff, E. et al, 2006). In this project, implicit information and social influence are handled by agents using conversational schemas. The implementation is based on Java and Jack agent toolkit. This paper is not very much related to our research but it addresses a part of the model. It does not talk about complete solution and architecture of social profiling model. Another agent based approach has been used in social computing that discusses socio-affective agent model (Tsung-Chuan Huang et al, 2003) to carry diagnostic reasoning development of domains with composite knowledge. This model provides semantic interoperability through web ontology language.

To interact with users and provide necessary services to avoid expensive and powerful mobile devices an agent based social model (Agent and Profiling Management System) was developed (Daniel Ramirez-Cano and Jeremy Pitt, 2006). In social networks a opinion formation model was proposed where confidence of dynamic nature in agent-mediated social networks was embedded (Toshihiko Yamakami, 2007). According to the model agents can change their opinions through learning from rest sources which bring them nearer to what they inclined to believe. Again it is a component of social computing that we have developed in our model to provide decision making property to the agent. To track user behavior regularity measures was proposed to examine patterns in the following dimensions: day count, day-of-week, time zone and time zone with day-of-week with three months transition patterns (Marcus J. Huber, 2007). This model lacks the autonomy that we have implemented in our model. Autonomy provides security to avoid external influence or social integrity (J. Doran, 1998). Agent's overall autonomy can be determined in many ways. One of the influencing ways is uncomplicated weighted computation of the social integrity and social dependency values with weighting based upon domain and agent-specified priorities. On the other side according to the theory of cognitive science there is strong relationship between agent-level and society-level phenomena (Anton Nijholt, 2003). The project of CASA (Computers Are Social Actors) was also developed to describe how the need of humans to build social relationships is anticipated through intelligence technology (Mark d'Inverno and Michael Luck, 2000). Agents are needed to develop such strong relationships. Social relations arise from the social action and mind of an individual (J. Ponce et al, 2006). Social environment is modeled by sociological agents.

The models, discussed above, mostly talk about interaction and cognitive issues. Less attention is paid to software agents that would create effective social interaction and build societies autonomously. The social computing through agents needs more attention to build societies those are very beneficial in making histories and develop communities.

While behavior is being analyzed the software agent would play key role because of its pro-activeness and other characteristics, discussed previously. We will design agent based user-behavior monitoring framework to identify the actual personality of user to avoid any kind of future deception. The developed models do not follow agent standardization; hence global community where agents may communicate each other and share their views in heterogeneous environment cannot be achieved. We will develop agent societies in the environment where agents can easily talk to each other without worrying about underlying platform.

4. Performance Evaluation Methodology and Standard Dataset:

It is very much necessary to take a valid dataset to assess whether the developed framework producing

right or wrong results. In following sections some standard dataset and evaluation methods have been investigated. Many efforts have been done to generate datasets capable of providing a challenging yet realistic platform for testing. Ponce (Caltech, 2003) presents the major issues in current datasets and provides suggestions for improvements. A list of commonly available dataset as well as complex, large scale category dataset is publicly provided. Little data exists today specific to the insider threat. Even what data exists, is specific to the organizations which prevent their dissemination freely to the research community. Datasets specific to the insider threat are merely available. Furthermore, the depth and breadth of such data is inconsistent. Reference control data is needed for evaluating research progress and results. Such data is essential for developing and validating insider threat models. The only insider threat dataset available is under development at Defense Security Research Center (DSRC) (S. Agarwal and D. Roth, 2002).

In order to analyze whether the results are correct or not, it is necessary to compare them with the ground truth data. While for the insider threat monitoring the best judge is how realistic the framework results are; there are certain well-defined and commonly used evaluation criteria in the literature. A brief overview is given below.

Generally, an evaluation can have one of the four values: True Positive, False Positive, True Negative and False Negative. Here “true” predictions imply a correct identification. For example, if an acceptable behavior is categorized as acceptable then it is a true positive; similarly if the system signals that there is no acceptable behavior and that is actually not acceptable then this is a true negative. False negative and false positive are errors, where false positive implies the rejecting a hypothesis that should have been accepted and false negative means the error of accepting a hypothesis that should have been rejected. Using these values, (Opelt, 2006) two performance measures were suggested: Recall Precision Curve (RPC) and Receiver Operating Characteristic (ROC) curve. True Positive Rate and False Positive Rate are the beneficial measurements. In ROC, TPR is plotted against FPR and a numerical value is extracted. According to Opelt (Paulo C. et al, 2000) Recall is simply TPR. Related concept is elaborated by other researchers in their own way. Paulo proposes that the outcome of the experiments of human and computer are always stated by two types of the probabilities, i.e. Probability of Detection (PD) and Probability of False Alarms (PFA). If threat behavior is correctly detected then it will become PD whereas a PFA occurs in the way that the user is identified to be a threat while case is not in that way and actually user is normal.

5. Conclusion:

The literature survey of three different and important domains has been done in this paper. In Agent Framework Standardization a comparison of two major framework standards shows that both standards are failed to focus all aspects especially security to the agent platforms. Therefore a new standard has been proposed to address the issues raised during research. Vulnerability Assessment especially insider threat is a major cause of financial and credibility loss of an organization, therefore an agent based model to avoid insider threat has been proposed. The Social profiling model proposed to keep the profiles of the internet users to trace their activities. It has been seen that software agents are still not well known and used in daily life applications therefore lot of implementation issues are there. After a detailed survey the development of an agent framework to detect and avoid insider threat is proposed and the vulnerability assessment and social profiling are addressed as the milestones towards it. In order to evaluate performance of the proposed framework the literature regarding standard dataset and appropriate performance measuring methodology have also been discussed.

REFERENCES

Agent Builder, 2000. “The agentbuilder user guide”, San Diego, California: Reticular Systems.

Anton Nijholt, 2003. “Disappearing Computers, Social Actors and Embodied Agents”, Second International Conference on Cyber worlds, pp: 128.

Alfredo Sánchez, J., 1997. “A Taxonomy of Agents” - Technical Report ICT-97-1 ICT Interactive and Cooperative Technologies Lab Universidad de las Américas-Puebla Department of Computer Systems Engineering A. P. 100 Cholula, Puebla 72820 México.

Agarwal, S. and D. Roth, 2002. “Learning a Sparse Representation for Object Detection”, in Proceedings of European Conference on Computer Vision, pp: 113-127.

Boff, E., Santos, 2006. “Social Agents to Improve Collaboration on an Educational Portal”, Sixth International Conference on Advanced Learning Technologies, pp: 896 - 900.

Caltech, 2003. Dataset Archive available at http://www.vision.caltech.edu/html_files/archive.

Christos Georgousopoulos and Omer F. Rana, 2002. “An approach to conforming a MAS into a FIPA-compliant system”, AAMAS’02, Bologna, Italy.

Daniel Ramirez-Cano, Jeremy Pitt, 2006. “Follow the Leader: Profiling Agents in an Opinion Formation Model of Dynamic Confidence and Individual Mind-Sets”, IEEE/WIC/ACM International Conference on Intelligent Agent Technology, (IAT'06), pp: 660-667.

Danny, B. Lange and Mitsuru Oshima, 1997. “A Security Model for Aglets”, IEEE Internet Computing, 1(4).

Doran, J., 1998. “Social Simulation, Agents and Artificial Societies”, Third International Conference on Multi Agent Systems, (ICMAS'98), pp: 4.

Etzioni, O. and D.S. Weld, 2007. “Intelligent Agents on the Internet: Fact, Fiction, and Forecast”, IEEE Expert, 10(4): 44-49.

Frank Manola, 1998. “Agent Standards Overview”, Object Services and Consulting, OBJS Technical Note.

Franklin, S. and A. Graesser, 1996. “Is It an Agent or Just a Program? Taxonomy for Autonomous Agents”, appeared in Proceedings of the Third International Workshop on Agent Theories, Architectures, and Language, New York: Springer-Verlag.

Gilbert, Aparicio, 1995. “The Role of Intelligent Agents in the Information Infrastructure”, IBM, United States.

Hyacinth S. Nwana, 1996. “Software agents: An Overview”, Knowledge Engineering Review, 11(3): 1-40. at Cambridge University Press.

Jamal Bentahar, Karim Bouzoubaa and Bernard Moulin, 2006. “A Computational Framework for Human/Agent Communication Using Argumentation, Implicit Information, and Social Influence”, Proceedings of the IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology, pp: 372-377.

Johansen, D., R. van Renesse and F. Schneider, 2000. “An introduction to the TACOMA distributed system”, Tech. Rep. 95-23, Institute of Mathematical and Physical Sciences, Department of Computer Science, University of Tromsø, Norway.

Jennings, N.R. and M. Wooldridge, 1998. “Applications of Agent Technology” Agent Technology: Foundations, Applications, and Markets. Springer-Verlag.

Katia P. Sycara, 1998. “Multiagent Systems”, AI magazine, 19(2) Intelligent Agents.

Lind, J., 2001. “Iterative software engineering for multiagent systems”, The MASSIVE Method - Lecture notes in Artificial Intelligence, Springer-Verlag, Berlin.

Maes and Pattie, 2005. “Designing Autonomous Agents”, Cambridge, MA: MIT Press, pp: 102-110.

Marcus J. Huber, 2007. “Agent Autonomy: Social Integrity and Social Independence”, International Conference on Information Technology (ITNG'07), pp: 282-290.

Menelaos, Perdikeas and S. Venieris, 1999. “An Evaluation Study of Mobile Agent Technology: Standardization, Implementation and Evolution”, Proceedings of the IEEE International Conference on Multimedia Computing and Systems, 2: 287-300.

Michael Pazzani and Daniel Billsus, 2004. “Learning and Revising User Profiles: The Identification of Interesting Websites”, Journal of Machine Learning, 27: 313-331.

Opelt, 2006. “Generic Object Recognition”, PhD thesis, Institute of Measurement and Measurement Signal Processing, Graz University of Technology, Austria.

Paulo, C., 2000. “DTB Project: A Behavioral Model for Detecting Insider Threats”, A project of Advanced Research and Development Activity (ARDA), supported by US Navy.

Pikoulas and J. Buchanan, 2002. “An intelligent agent security intrusion system”, Ninth Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, pp: 94 - 99.

Stuart Russell and Peter Norvig, 2002. “Artificial Intelligence: A Modern Approach”, Englewood Cliffs, NJ: Prentice Hall, pp: 375-410.

Tobias, M.W., 2000. "Locks, Safes and Security", 2nd Edition, Charles Thomas Publisher, Ltd, Springfield, IL, USA.

Toshihiko Yamakami, 2007. “Classification of Mobile Internet User Behaviors using Qualitative Transition Patterns”, International Conference on Information Technology, (ITNG'07), pp: 890-892.

Tosic, Predrag and Gul Agha, 2004. "Towards a Hierarchical Taxonomy of Autonomous Agents," Proceedings of IEEE International Conference on Systems, Man and Cybernetics (IEEE-SMC'04), The Hague, The Netherlands.

Tsung-Chuan Huang and Chu-Sing Yang, 2003. “An Agent and Profile Management System for Mobile Users and Service Providers”, 17th International Conference on Advanced Information Networking and Applications (AINA'03), pp: 574-280.

Wang Zeng-quan, Wang Hui-qiang and Zhang Rui-jie, 2006. ”Research and Design on Intelligent Agent Intrusion Response System”, International Conference on Intelligent Agents, Web Technologies and Internet Commerce, 3: 231 - 233.

Zhong Zhang and McCalley, 2004. “Multiagent System solutions for distributed computing, communications, and data integration needs in the power industry”, Power Engineering Society General Meeting, IEEE, 1: 45 - 49.