International Journal of Advanced Trends in Computer Applications
www.ijatca.com

Modeling Agent-Based Network Monitoring Application using UML

[1]Ojieabu C. E., [2]John-Otumu A. M., [3]Oshoiribhor E. O.
[1]Dept. of Elect/Elect
Ambrose Alli University Ekpoma, Nigeria
[2]ICT Directorate
Ambrose Alli University Ekpoma, Nigeria
[3]Dept. of Computer Science
Ambrose Alli University Ekpoma, Nigeria

Abstract: *Network monitoring applications are becoming critical systemsand increasingly complex to manage. To help manage this complexity, the application needs to be modeled properly in order to understand its challenges and possible solutions to adopt. Unified Modeling Language (UML) is the standard language for modeling software intensive systems like network monitor applications. In this paper, we used the use-case, sequence, and state chart diagrams to model our design user interface interactions, internal mechanisms behavior and application state. UML is recommended for software architects, application developers and researchers in the field of software engineering for vital technical documentations.*

Keywords: *Agent-based, Network monitor, UML, Modeling.*

I. INTRODUCTION

Nowadays network monitoring applications are used in a wide variety of environments, including Local Area Networks (LAN), Wireless Local Area Network (WLAN), Enterprise LAN, Distributed LAN Environment, Wide Area Network (WAN) to mention just a few. Before now, monitoring network environment with a view to resolving problems, and ensuring optimal performance and efficiency normally involve the physical movement of the network administrator from one computer system to another [1, 2], but according to [3] the manual monitoring of a network environment by a system or network administrator(s) can be a very huge task and cannot satisfy the requirements of the modern complex network system.

A plethora of techniques were reported in literatures on agent-based software approaches for network management using static agents which are stationary and would not need to move from one node to another in order to execute its objective [3], mobile agent network management application that has the ability to move or migrate from one node to another in a network environment in order to perform a given task on behalf of the administrator [1], and fault management in computer network[4].

In this paper, we address the capabilities of UML in expressing its modeling requirements in agent-based network monitoring applications. Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of object-oriented software engineering to model the multi-agent based network monitoring application. UML includes a set of graphic notation techniques which creates visual models of objects-oriented software intensive systems.

UML can be used to model different kinds of systems: software systems, hardware systems, and real-world systems. UML offers nine diagrams in which to model systems:

i. Use Case diagram for modeling the business processes
ii. Sequence diagram for modeling message passing between objects
iii. Collaboration diagram for modeling object interactions
iv. State diagram for modeling the behavior of objects in the system
v. Activity diagram for modeling the behavior of Use Cases, objects, or operations
vi. Class diagram for modeling the static structure of classes in the system

vii. Object diagram for modeling the static structure of objects in the system
viii. Component diagram for modeling components
ix. Deployment diagram for modeling distribution of the system.

We also reviewed some application areas of UML in modeling different systems. According to [5] UML was used to model profiles for real-time systems and their applications. Sequence, class, collaboration and state diagrams were used in modeling different mechanisms, while [6] used use-case, class, state and collaboration UML diagrams in modeling real-time embedded systems. There are also some related works reported on application of UML to embedded systems design by[7], [8], [9], [10].

II. METHODOLOGY

Models can help us understand the context of application system by simplifying some of the details. The right choice of what to model has an enormous effect on the understanding of the problem and the shape of the solution [11]. Basically, we used UML approach to model the agent network monitoring application architectures in the following areas:

i. Admin interaction with user interface [Use Case Diagram]
ii. Anomaly intrusion detection mechanism [Sequence Diagram]
iii. Fault detection mechanism [Sequence Diagram]
iv. Shutting down mechanism [Sequence Diagram]
v. Agent transition [State Diagram]

The use-case, sequence, and state machine diagrams were carefully selected in modeling the agent network monitoring application because they provides wide array of diagrams for analysis and design modeling at both the system and the software level.

Use Case Diagram

The use case diagram describes what a system does from the standpoint of an external observer. The emphasis is on what a system does rather than how is does it. Use case diagrams are closely connected to scenarios. Figures 1 and 2 are practical scenarios of the features a network administrator can interact with using the proposed network monitoring application.

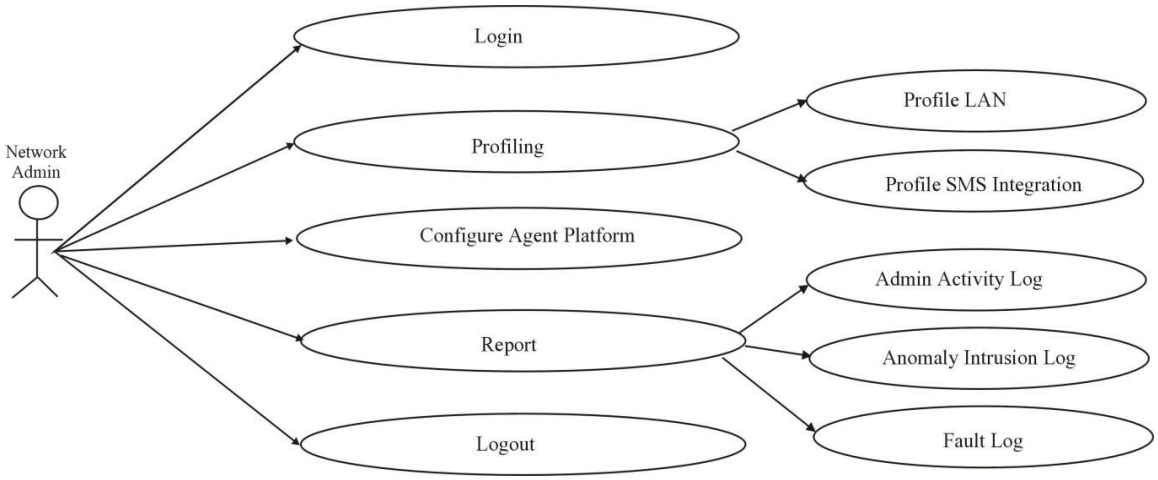

Figure 1: Modeling network administrator's interaction with the system interface features

Figure 1 show the network monitor system features like login, profile nodes in a LAN, configure agents platform, reports, and logout modules the administrator can interact with in order to properly administer and manage the network environment. The profiling module has two sub-modules i.e. profiling LAN and profiling SMS integration, while the report module also has three sub-modules i.e. admin activity log, anomaly intrusion log and the fault log. Note: the configure agent platform has eight sub-modules though not represented in the diagram (see figure 2).

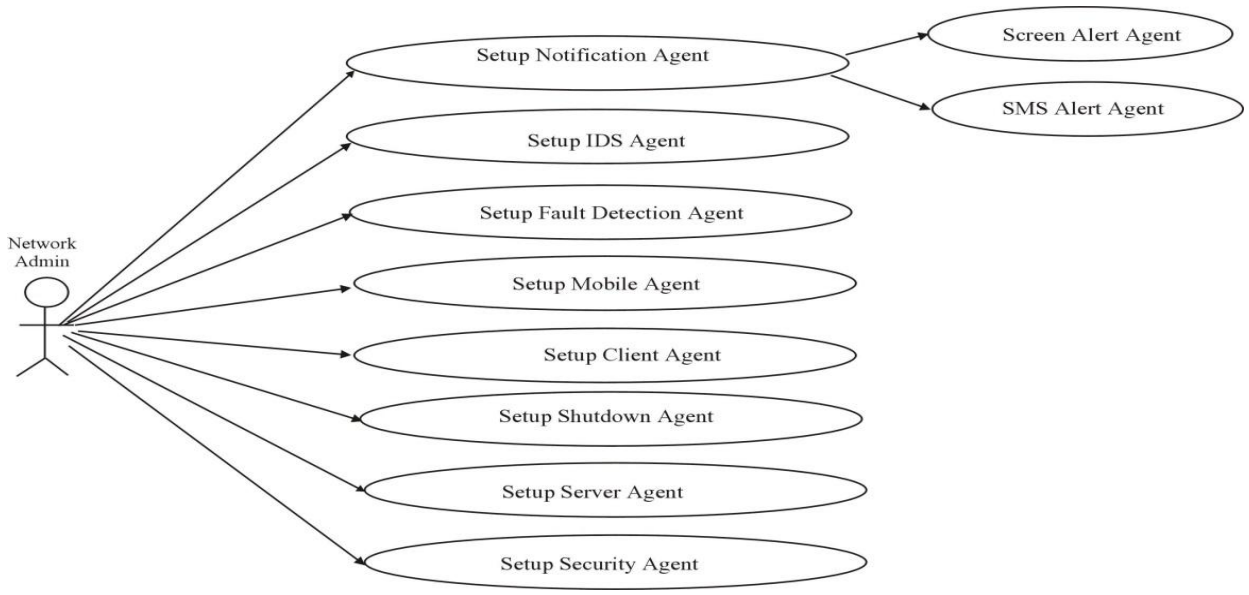

Figure 2: Modeling network administrator's interaction with the configure agent platform menu

Figure 2 shows the interaction between the network administrator and the network monitoring application in order to setup and activate the various agents for coordination /communication and performance of different desired task.

Sequence Diagram

A sequence diagram is an interaction diagram that details how operations are carried out, what messages are sent and when it is sent. It shows the interaction between objects or class in order to get a particular task or process done. Sequence diagrams are organized according to time. The time progresses as you go down the operations. The objects involved in the operation are listed from left to right according to when they take part in the message sequence.

Here, we used the sequence diagram to model different interactions within the internal mechanism of the multi-agent network monitoring application.

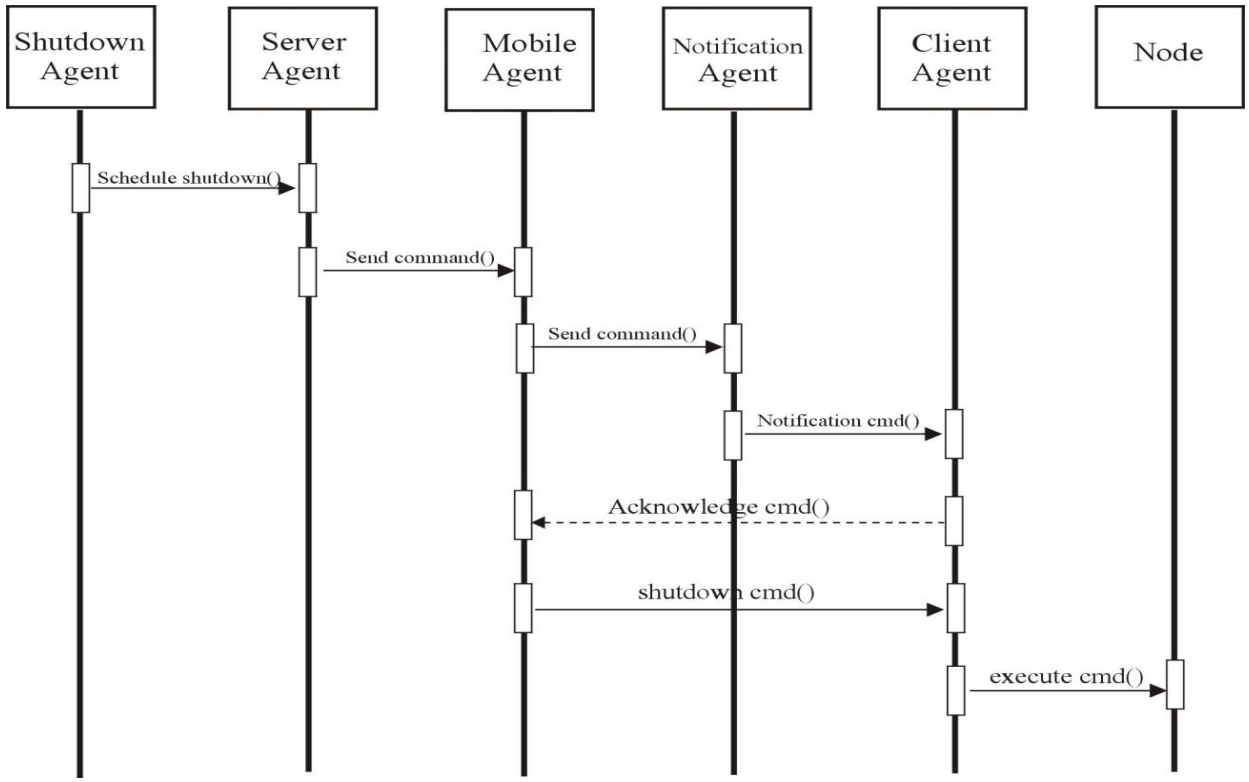

Figure 3: Modeling interaction between agents for shutting down nodes

Figure 3 depicts the interaction between five agents in order to shut down nodes in the network environment. The agents involved in the shutdown operation are shutdown agent, server agent, mobile agent, notification agent and the client agent. The shutdown agent schedule the shutdown time and send command to the server agent. The server agent receives the command operation and also forwards it to mobile agent. The mobile agent receives the shutdown command and holds operation on it until the scheduled time to execute; but rather send control to the notification agent indicating the scheduled time for shutting down the nodes on the network. The notification agent also passes command to the client agent resident in each node on the network. The client agent sends an acknowledgment to the mobile agent indicating its readiness to carry out the shutdown operation. On exactly the scheduled time the mobile agent executes operations and passes command to the client agent to shutdown the node using the remote object command.

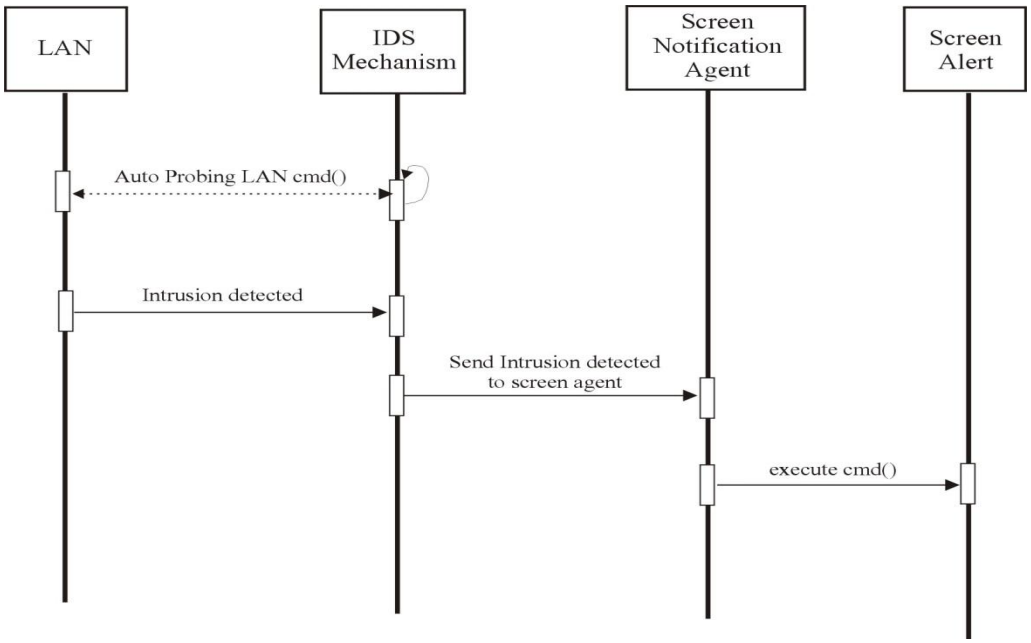

Figure 4: Modeling intrusion detection mechanism to screen alert system

Figure 4 depicts the process flow and the agents responsible for detecting and reporting anomaly node intrusion from the computer network environment to the screen alert system. The Intrusion Detection System (IDS) agent constantly probes the network, gets and analyzes the traffic patterns for anomalies. If any anomaly is detected by the agent mechanism it immediately signals the notification agent which in turn sends control to the screen alert system for a near real-time reporting based on the computer processing power.

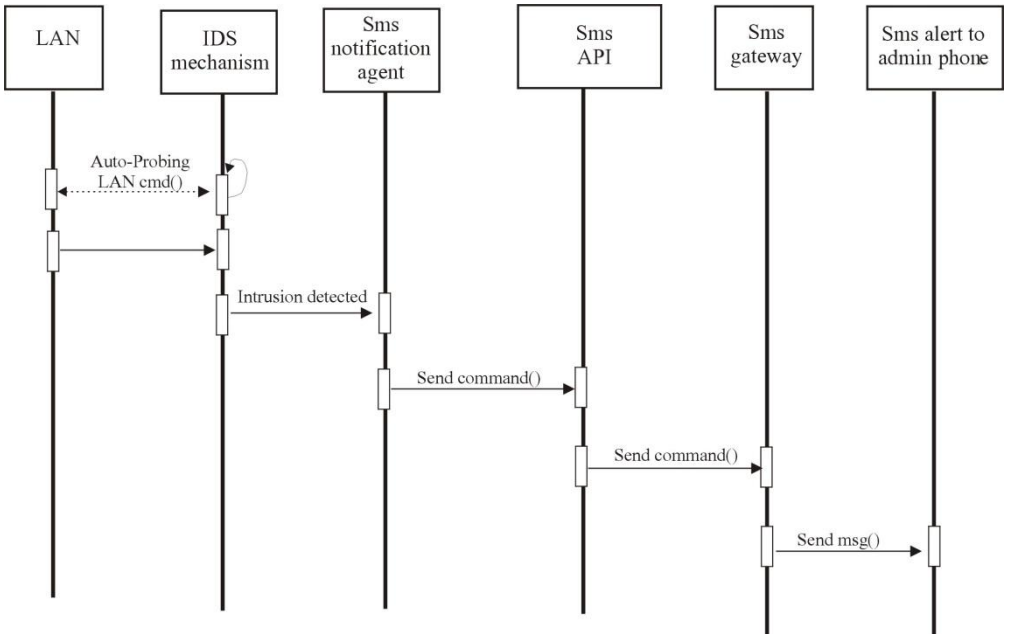

Figure 5: Modeling intrusion detection mechanism to sms alert system

Figure 5 depicts the process flow and the agents responsible for detecting and reporting anomaly node intrusion from the computer network environment to the short message service (SMS) alert system. The Intrusion Detection System (IDS) agent constantly probes the network, gets and analyzes the traffic patterns for anomalies. If any anomaly is detected by the intrusion agent mechanism it immediately signals the sms notification agent which in turn sends control to the SMS Application Programming Interface (API) from there to the SMS gateway and finally to the network administrator's mobile phone as text message alert.

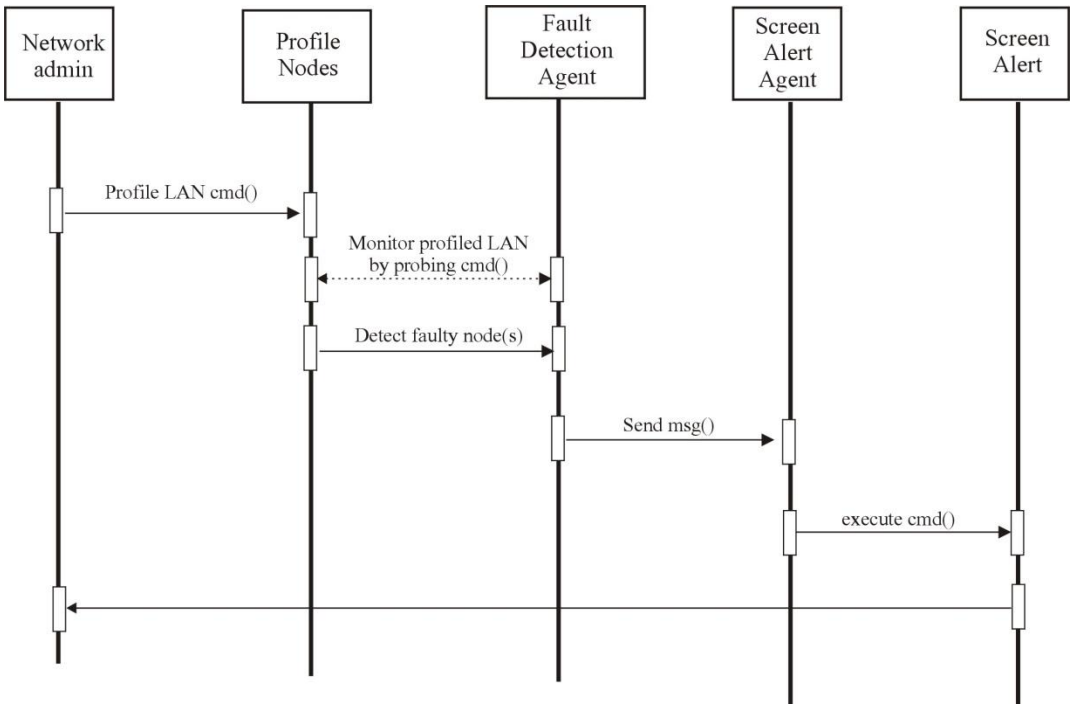

Figure 6: Modeling fault detection mechanism to screen alert system

Figure 6 depicts the process flow and the agents responsible for detecting and reporting faulty nodes in the computer network to the screen alert system. The network administrator profiles all available nodes on the network in the first instance. The fault detection agent constantly probes the network, gets and analyzes the traffic patterns for from each node to know its status i.e. if active or inactive. If any node is detected by the fault agent mechanism to be inactive or down, the mechanism immediately signals the screen alert agent which in turn sends control message to the screen in near real-time.

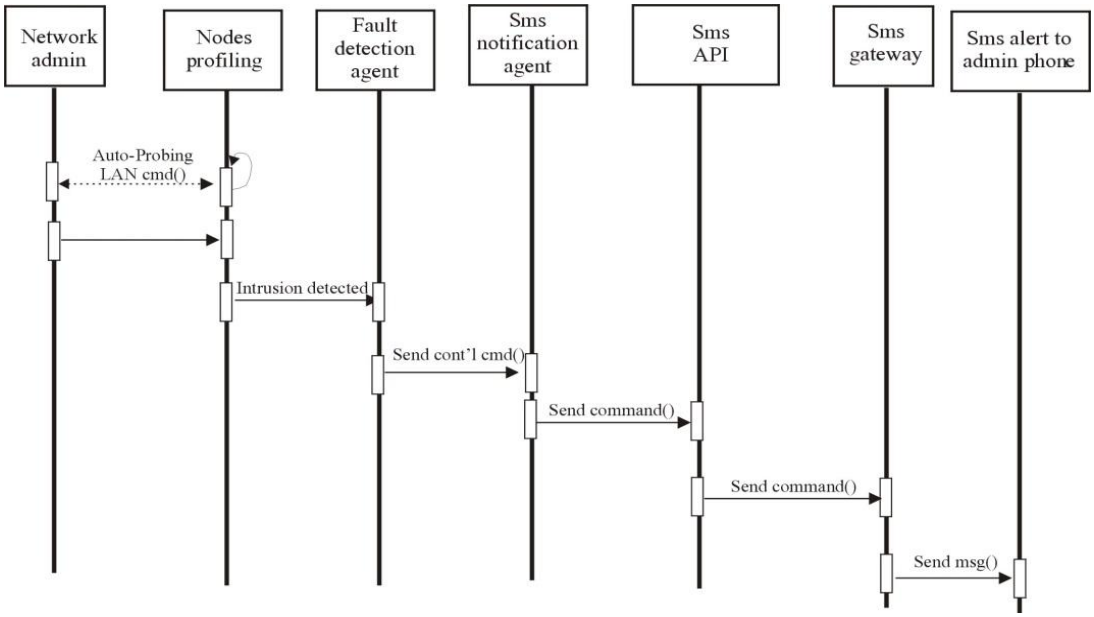

Figure 7: Modeling fault detection mechanism to sms alert system

Figure 7 depicts the agent fault detection mechanism responsible for detecting and reporting faulty node on the computer network to the short message service (SMS) alert system. The fault detection agent constantly probes the network in order to get and analyze traffic patterns from profiled nodes on the network. It constantly examines each node status for its active or inactive state. Once any node is seen or detected to be inactive, the mechanism signals the sms notification agent which also passes control to the sms API, which in turn passes the fault alert message to the network administrator's mobile phone as text message through the sms gateway.

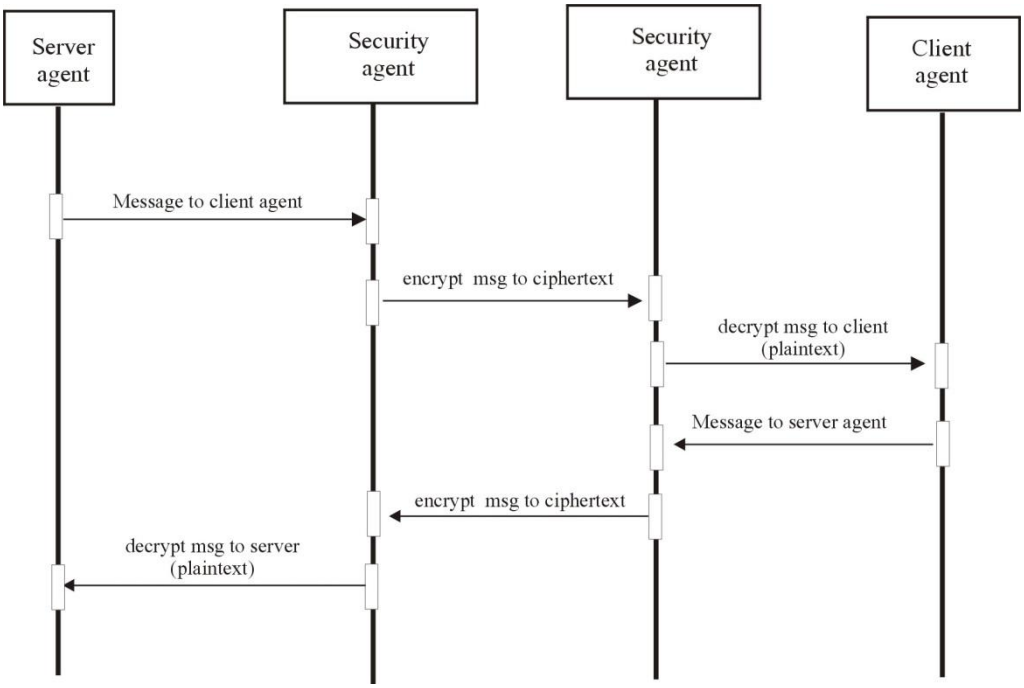

Figure 8: Secure end-to-end encryption between server and client agent

Figure 8 shows the interaction between the server, security and client agents in order to have a secured end-to-end communication process using Data Encryption Standard (DES) algorithm. The security agent uses the DES algorithm to mathematically transform data from plaintext to cipher text and back to plaintext again with the concept of encryption and decryption. This secured communication process is necessary in order to avoid theft or eavesdrop of data by other malicious agents or hacker. Each node data profiled (node name, ip address, mac address, system name) is always being synchronized between the server and client agents for update information from both ends since a network environment is dynamic. Malicious agents can position themselves between the server and client agent to steal data during this update process for their personal benefits. When this occurs all they can see will be scrambled data that might be useless to them.

State chart diagram

A state can be referred to as a condition in which an object can be at a particular point during its lifetime for some finite period of time [12]. State diagrams can also describe all the possible states a particular object can get into and how the objects state can change as a result of an external occurrence of events that reach the object[13].

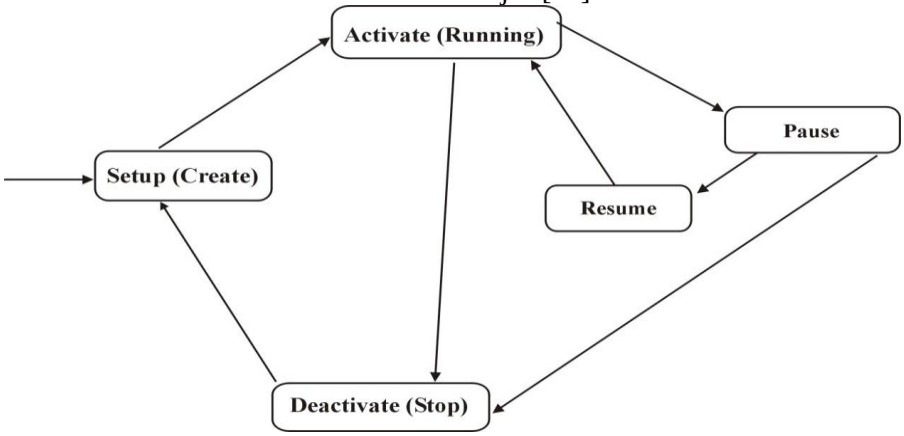

Figure 9: Agent state machine diagram

Figure 9 depicts our proposed multi-agent state chart diagram transiting between five states. The states are setup (create), activate (running), deactivate (stop), pause, and resume. Here, the agent can be setup or created, next is to activate it to running state, from running state the agent can be deactivated / stopped or it can rather be paused temporarily from running. From the paused state, the agent can resume back to running state or it can be deactivated or stopped.

III. DISCUSSION

In this section, we discuss some research issues related to application of UML in system modeling. From the works of previous literature reviewed in [5] and [6], we can comfortably compare our design model to the work done by those researchers.

We used sequence and statechart diagrams to model different internal mechanisms like anomaly intrusion, fault mechanism etc in the agent-based network monitoring application, while [5] also used sequence, state and other UML diagrams like collaboration and class to model their real-time system application. We also used use-case to model administrator's interaction with the agent-based network monitor application user interface; which is also an expressive modeling support for [6] alongside side with collaboration, class and state diagrams.

CONCLUSION

Although this paper provides only a brief introduction to Unified Modeling Language, we also tried to explain the various forms or types of models the UML can be used for. Modeling applications using UML will help software project team as well as researchers in the field of software engineering understand application problems and software behavior more with respect to design and development. Though there are several special software tools that could aid to integrate UML diagrams into software development process, but application packages like Coreldraw, paint or any other graphic application can also help much in designing simple or complex models or even without any automated tools, you can also use pencil or pen to sketch or draw simple UML diagrams and still achieve same goal or benefits.

REFERENCES

[1]. Imianvan, A. A. (2009). *Development of Mobile Agent for Evaluating the Use of Bandwidth in a Computer Network*, PhD Thesis, Department of Computer Science, Federal University of Technology, Akure. In (Akinyokun, O. C., Ekuewa, J. B., and Arekete, S. A., 2014) *Development of agent-based system for monitoring software resources in a network environment*, Artificial Intelligence Research, Published by Sciedu Press, Vol. 3, No. 3

[2]. Arekete, S. A. (2013). *Development of a Mobile Agent for Monitoring and Evaluation of Activities of Users in a Network Environment*, PhD Thesis, Department of Computer Science, Federal University of Technology, Akure, In (Akinyokun, O. C., Ekuewa, J. B., and Arekete, S. A., 2014) *Development of agent-based system for monitoring software resources in a network environment*, Artificial Intelligence Research, Published by Sciedu Press, Vol. 3, No. 3

[3]. Akinyokun, O. C., Ekuewa, J. B., and Arekete, S. A. (2014). *Development of agent-based system for monitoring software resources in a network environment*, Artificial Intelligence Research, Published by Sciedu Press, Vol. 3, No. 3

[4]. Jailani, N., & Patel, A. (1998). FMS: A computer network fault management system based on the OSI standards. *Malaysian Journal of Computer Science*, *11*(1), 22-31.

[5]. Gherbi, A., and Khendek, F. (2006): UML Profiles forReal-Time Systems and their Applications, Journal of Object Technology, vol. 5, no. 4,May–June 2006, pages 149–169,http://www.jot.fm/ issues/issues 2006 05/article5

[6]. Kaur, A. and Arora, R. (2012). Application of UML in Real-Time Embedded Systems. International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.2, March 2012

[7]. Riccobene, E., Scandurra, P.,Rosti, A., and Bocchio, S. (2005). "A SoC Design Methodology Involving aUML 2.0 Profile for SystemC", In Proceedings Design, Automation and Test in Europe.

[8]. Schattkowsky, T. and Muller, W. (2004). "Model-Based Design of Embedded Systems", Proceedings of 7th IEEE International. Symposium on Object-Oriented Real-Time Distributed Computing.

[9]. Zakaria, N. A., Kimura, M., Matsumoto, N., and Yoshida, N. (2009). "Stepwise Refinement in Executable-UMLfor Embedded System Design: A Preliminary Study", World Academy of Science, Engineering and Technology.

[10]. Wang, G. (2009). "Modeling C-based Embedded System using UML Design" Proceedings of IEEE International Conference on Mechatronics and Automation.

[11]. Conallen, J. (1999).Modeling Web Application Architectures with UML, Rational Software. Communications of theACM. volume 42, number 10.

[12]. Scott, K. (2001). UML Explained. Boston, Massachusetts, Addison-Wesley.

[13]. Fowler, M. (2000). UML Distilled. Reading, Massachusetts, Addison Wesley.

ABOUT THE AUTHORS

Engr. (Dr.) Ojieabu Clement Eghosaholds a Ph.D in Communication Engineering, M.Eng in Electronic & Telecommunication Engineering and B.Eng in Electrical/Electronic Engineering. He is an Associate Professor of Communication Engineering at the

Department of Electrical/Electronic Engineering, Ambrose Alli University, Ekpoma. He is a registered engineer with the Council for the Regulation of Engineering in Nigeria (COREN). His research interest includes Data communication and security, intelligent systems and satellite communication systems. He has published over 30 articles in both local/international journals and conference proceedings. He can be reached at ***bishopeghosa@yahoo.ca.***

John-OtumuAdetokunboMacGregoris a Ph.Dstudent at the Department of Computer Science, Ebonyi State University, Abakaliki, Nigeria. He obtained his M.Sc(Info Tech) from National Open University of Nigeria and M.Sc (Computer Science) from Ambrose Alli University, Ekpoma, Nigeria. He is a Senior Technologist in the Directorate of Information and Communication Technology, Ambrose Alli University, Ekpoma, Nigeria. He is a registered member of the Nigeria Computer Society (NCS), Institute of Software Practitioners of Nigeria (ISPON) and a Chartered Information Technology Professional registered with the Computer Professionals Registration Council of Nigeria (CPN). His research interest includes computer communication systems, agent computing and multi-agent based systems, network securityand soft computing. He has published over 17 articles in both local and international Journals.He can be reached at ***macgregor.otumu@gmail.com.***

Dr. Oshoiribhor Emmanuel Osazeholds a Ph.D in Computer Science from Ambrose Alli University, Ekpoma, Nigeria, M.Sc., and B.Sc. (Hons) Computer Science from University of Benin, Nigeria. He is a Lecturer in the Department of Computer Science, Ambrose Alli University, Ekpoma, Nigeria. He isa registered member of Nigeria Computer Society (NCS). His research interest includes Data Mining, Artificial Intelligence, and Software Engineering He has published over 16 articles in both local and international Journals. He can be reached at ***emmaoshor2001@gmail.com***.