Agent-Based Faults Monitoring in Automatic Teller Machines.

[1]Bashir Sulaimon Adebayo, [2]Mohammed Idris Kolo
[1&2] Department of Computer Science
School of Information and Communication Technology
Federal University of Technology
Minna, Niger State, Nigeria
bashirsulaimon@futminna.edu.ng[1] kolos_2006@yahoo.com[2]

Abdulsalam Sulaiman O.
Department Of Computer, Library And Information Science
College Of Information and Communication Technology
Kwara State University, Malete, Kwara State, Nigeria
[3]*sulepy@yahoo.com*

ABSTRACT

Automated Teller Machine (ATM) has gained widespread acceptance as a convenient medium to facilitate financial transaction without need for human agent. However, ATM deployers are facing challenges in maximizing the uptime of their ATMs as a result of wide gap in fault detection, notification and correction of the ATMs. One way to ameliorate this situation is through intelligent monitoring of ATM by resident software agents that monitor the device real time and report faulty components real time to facilitate quick response. We proposed an architecture for rule-based, intelligent agent based monitoring and management of ATMs. Agents are used to perform remote monitoring on the ATMs and control function such software maintenance. Such agents can detect basic events or correlate existing events that are stored in a database to detect faults. A system administrator can securely modify the monitoring policies and control functions of agents. The framework presented here includes software fault monitor, hardware fault monitor and transaction monitor. A set of utility support agents: caller agent and log agent are used to alert network operator and log error and transaction information in a database respectively. at-1, stuck-at-0 faults in digital circuits validate the point that faulty circuits dissipates more and hence draw more power.

Key words: Automated Teller Machine (ATM), Intelligent Agents, Mobile Agents, Event Monitoring.

1. INTRODUCTION

The emergence of the application of Information communication technology in the operation of financial institutions has greatly enhanced their operation and has brought about electronic means of conducting financial transactions. The electronic payment system involves the provision of payment services and transfers through such devices as telephones, computers, the internet, ATMs, and smart cards [1,2]. As a paperless system of making payment, electronic payment systems have the advantage of enabling transactions to be processed quickly and more cheaply and also offer a much more convenient method of effecting settlement of transactions [3]. The points of interaction include point-of sale (POS) terminals, automated teller machine (ATM), etc.

ATM in particular is enjoying the widest patronage as alternative means of conducting financial transaction without human agents. It is a source of huge revenues for banks and other ATM support service providers. Banks and other ATM deployers are therefore paying more attention to deploy more ATMs at strategic locations to increase their users' base and also to earn more transaction service fees while also enjoy inflow of deposits from ATMs that support deposit of cash [4].

As we move towards cashless society, ATM will continue to play a vital role in paperless financial transactions such as payment of utility bills like electricity, water, phones, etc, electronic cash transfer, cash deposit, checking of account balances and lot of new and innovative services. The need for low-value transaction also means that ATM will continue to facilitate dispensing of low cash denominations for payment of low-value transactions especially in the informal sector of the economy.

All these indicate the significance of ATMs in financial transactions. According to ATM Industry Association of Europe (ATMIA) [5], the number of ATM worldwide in 2007 was over 1.6million, and forecast by Retail Bank Research in 2008 indicates that this number will be over 2.5 million by 2013 [6]. The key challenge facing ATM deployers is ensuring customer satisfaction through continuous availability of ATM services 24/7 with a very minimal downtime. Monitoring multitude ATM devices both located in-premise and off-premise has become a challenging and tedious task for today's ATM device engineer as time delay to resolve fault is often longer than necessary. This often leads to customer dissatisfaction, loss of revenue and even a total loss of customers changing to bank with better ATM uptime.

To ameliorate this situation, we propose an agent based monitoring of ATM to enhance their uptime and thereby increase customer satisfaction. Agents are used to perform remote events monitoring on the ATMs and can perform control functions such software reset. Such agents can detect basic events or correlate existing events that are stored in a database to detect faults. A system administrator can securely modify the monitoring policies and control functions of its agents, or install new agents at a node. The framework presented here includes software events monitor, hardware events monitor and transaction monitor. A set of utility support agents: caller agent and log agent are used to alert network operator and log error and transaction information in a database respectively.

This paper is organized as follow, next section gives a review of related work, section 3 presents the details of the agent based monitoring architecture. Section 4 discusses the future work and conclusion follows in section 5.

2. RELATED WORK

Network management is a broad area that encompasses the provision of services that use varieties of tools, applications and devices to assist human agent in monitoring and maintaining networks [7]. The services provided include: Network monitoring which performs fault detection, reporting and repair. Network configuration management entails initializing network devices- such as host address and parameters configuration, service provisioning and adding and updating information about. Network Performance management and security monitoring performs accounting, performance and ensure security of the network against intrusion and attacks.

Intelligent Agent or software agent is a new paradigm emanating from Artificial Intelligence community, which connotes the development of software, or program that assists people and acts on their behalf and allow people to delegate work to them. A more technical definition given in [8] defines an agent as software object situated within an execution environment with the following mandatory properties: **-Reactive:** senses changes in the environment and acts according to those changes; **Autonomous**: has control over its own actions; **Goal driven:** is proactive; **Temporarily continuous:** is continuously executing. Agent may possess any of the following orthogonal properties: ***Communicative:*** able to communicate with other agents; ***Mobile:*** can travel from one host to another ***Learning:*** adapts in accordance with previous experience and ***Believable:*** appears believable to the end user.

Software agent paradigm has been applied to wide problem areas including but not limited to electronic commerce[9,10], personal assistance [11], distributed information retrieval[12,13], telecommunication network services, workflow applications and groupware, network management[14,15,16] etc. In [20], a survey of research efforts on the use of multidisciplinary modern techniques in distributed artificial intelligence, machine learning and neural networks for network management were carried out. Generally in computer networks including ATM systems; resource utilization, number of users and types of networks are dynamic factors, therefore faults and problems that can occur would vary in terms network scope, types, complexity and number of affected devices.

Authors in [20] opined that adapting self managing and self improving capabilities of artificial intelligence and computational Intelligence can proffer solution to these problems. Efforts in the application of Artificial Intelligence include: [22]: Application of autonomous intelligent agent with SNMP to Network management; [23]: Distributed Artificial Intelligence agents for network management framework of high speed LAN of linear topology using deterministic multiple access protocol and [24]: Application of traffic engineering system of intelligent agents for scalable, resilient and effective traffic management in MPLS network using time series analysis for forecasting of network load for optimal resource provisioning and automated traffic engineering.

In [21], the benefit of conventional network management was identified as flexibility of managing the whole network in a single place while the demerits include, information bottleneck, excessive processing at the management station and heavy usage of network bandwidth in communicating with all the managed stations. They proposed an agent based network monitoring system that utilized hybrid of agent and the traditional SNMP approach. They also proposed two agents travelling patterns- itinerary and broadcast model with their usage advantages and disadvantages. Their approach purely uses mobile agents, which also utilized network bandwidth to travel form central station to managed station to collect monitoring data. We use a combination of mobile and stationary agents to reduce bandwidth consumption in moving from one node to another.

Over the years, a number of approaches have been proposed to enhance the detection of ATM faults. Author in [26] established the negative effect of continuous downtime of ATM devices to customer loyalty and reputation of business and point out the need for remote management solution to complete existing device level add-ons. Author proposed ESQ’s remote monitoring agent for Operations Bridge (RMMA for OB) tool. The tool allow medium to medium- to large-sized ATM deployers’ to track a multitude of events at the machine level with a single resident device on the ATM .

In [17], a principal component analysis (PCA) method for detection of unexpected behaviours of ATM was proposed. The unexpected behaviours of specific ATM was detected using PCA models of ATMs joined in special ATM cluster. When the correlation models of the ATMs’ cluster had given large prediction error an unexpected behaviour of the specific ATM was declared. However, the inaccuracy of PCA models in [17] to the detection of unexpected behaviours in ATM was improved by [25] through non-linear correlation model for describing ATM’s daily money withdrawals.

Furthermore, a survey of ATM malfunctioning in selected states in Nigeria was presented in [18]. The survey captured four categories ATM faults grouped into: Delayed Cash, Trapped Card, over billing and cannot dispense cash scenario. The corresponding algorithm for resolving each category faults was proposed for the detection of the malfunctions. Our work in this direction focused on the design of a framework for detecting ATM faults using intelligent software agent paradigm.

3. METHOD AND DESIGN

In this section, we describe an action rule-based, intelligent software agent for ATM fault monitoring system architecture. The agents are resident on individual ATM and use the low-level CEN/XFS API to probe the hardware components of the ATM. Each hardware component as explained later has set of APIs to probe their state and monitor their health condition. The software components of the ATM are monitored through the Window Management Interface (WMI) to determine the state of the core software components on the system.

The physical architecture the of system is depicted in Figure 1 consisting of set of ATM devices connected to the central system management station which is host the central monitoring agent. The server also presents an interface for human operators to interact with the system and perform administrative functions such as system initialization, policy changes, maintenance, and updates.

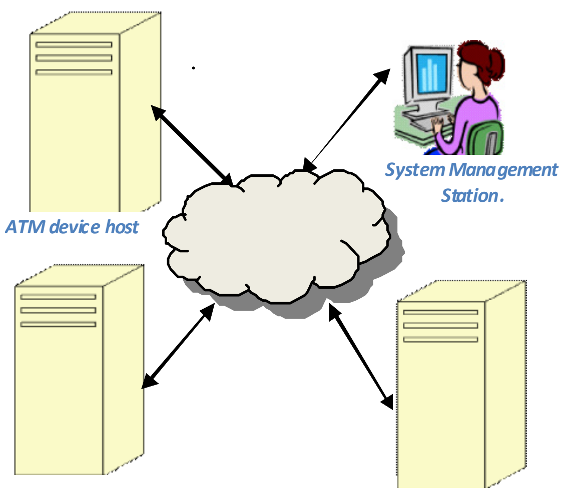

Fig 1: Physical Architecture of ATM Monitoring

3.1 Agent Structures on the Hosts

Each host has architectural layers as shown in figure 2 below. Each host in the monitoring network except the system management station has the architectural layers consisting of the hardware and software interface, the agent execution environment (the engine) and the monitoring agents. The agent execution environment is integrated between the operating environment such the Windows Operating System and the hardware and software monitoring agent.

This execution environment provides agent hosting services such as inter-agent communications, agent life-cycle management, mobility and other agent related functionalities desired by the agent framework.

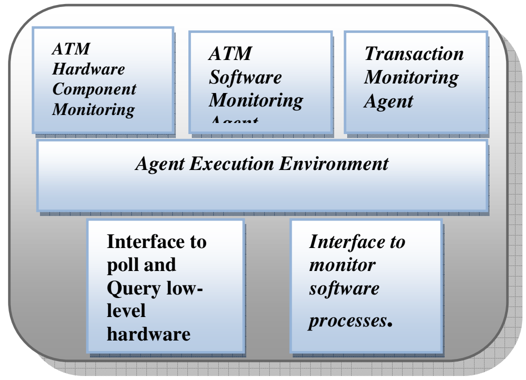

Figure 2 : Architectural Layers of the System

3.2 The Hardware and Software Interface

At the lowest layer of the architecture is the interface to the hardware and software environment. The ATM platform low-level device details will be provided through a set of standardized interface. The advent of industry standard interface like CEN/XFS standard [19] makes this function easier to perform. The CEN/XFS standard prescribes a general language **t**o speak with the hardware components of the Automated Teller Machine (ATM).

The CEN/XFS architecture is similar to a client-server model. The server knows how to drive a particular ATM hardware component. The server must publish a standard interface which is common for all similar hardware components. The client benefits from the services provided by the server (also called Service Provider or SP) and use its standardized interface to drive hardware components from different manufacturers with the same set of commands.

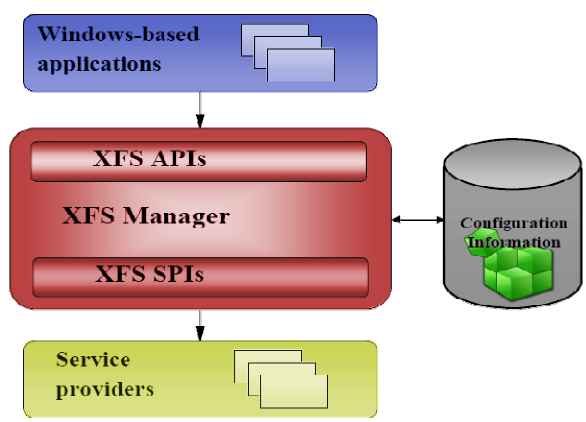

Fig 3: CEN/XFS Architecture
(Source: cen-xfs-atm-software.info)

The applications communicate with SPs, via the XFS Manager, using the API set, which is the common deliverable in all implementations of the XFS specification. It maps the specified API to the corresponding SP Interface (SPI), then routes this request to the appropriate SP.

The XFS Manager uses the configuration information to route the API call (made to a "logical service") to the proper SP entry point, and provides overall management of the XFS subsystem. The primary function of the SP (Service Provider) is to translate the generic XFS requests to the managed hardware component specific command. The ATM application uses the ATM hardware components through the XFS Manager and does not need to know anything about the hardware type and manufacturer. Supported device classes include: Printers and Scanners, (Ptr), Identification Card Units (Idc), Cash Dispensers (Cdm), Personal Identification Number Keypads (Pin), Check Readers and Scanners (Chk), Depository Units (Dep), Text Terminal Units (Ttu), Sensors and Indicators Units (Siu), Vendor Dependent Mode (Vdm), Cameras (Cam), Alarms (Alm), Card Embossing Units (Ceu), Cash-In Modules (Cim), Card Dispensers (Crd), Barcode Readers (Bcr), Item Processing Modules (Ipm).

Most modern ATMs are currently based on the Windows Operating system running different software. The Windows Management Interface (WMI) provides access to operating system data and ability to monitor software processes.

3.3 Hardware Monitoring Agent

Hardware monitoring agent is in charge of monitoring various hardware components such as card reader, cash dispensing mechanism, printer, speaker and the keypad of the ATM device. Different hardware monitoring agents monitor each hardware component of the ATM. The low-level hardware interface provides the agents with the mechanism of probing the device state and be able to report their state to the central monitoring agent on the management station. The structure of hardware monitor agent is shown in Figure4.

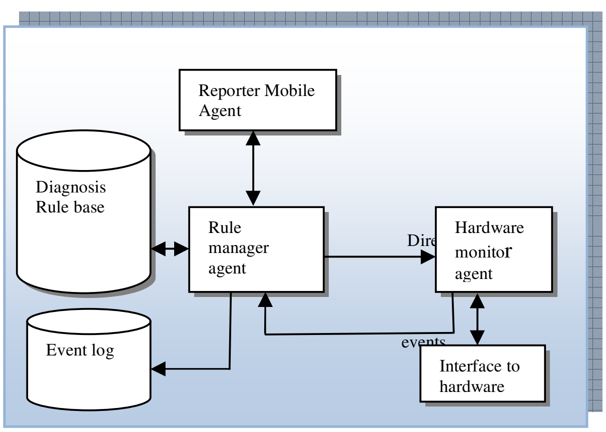

Fig 4: Hardware Monitor Component Architecture

The rule repository contains the rules that are used to diagnose and act appropriately. The statements in the rule are of the form: If eventOccur() then actionTaken() eventOccur ( ) will utilize CEN/XFS commands to probe the hardware components to determine the kind of problem that has occurred. For example, to determine the status of printer, the command WFS_INF_PTR_STATUS can be used.

The command returns the status of the printer as WFSPTRSTATUS and could assume different values based on the status of the device. The psedoocde of eventOccur( ) action rule procedure can be as follow:

```
eventOccur( ) {

 WFSPTRSTATUS =   WFS_INF_PTR_STATUS( );

If ( WFSPTRSTATUS = =
WFS_PTR_DEVFRAUDATTEMPT)
//The device is present but is inoperable
//because it has detected a fraud attempt.
actionTaken(activateReporterAgent(Management
Station));

else if ( WFSPTRSTATUS = =
WFS_PTR_DEVPOTENTIALFRAUD)
// The device has detected a potential fraud attempt
//and is capable of remaining in service.
actionTaken(shutdownAndNotify( ));

else if (WFSPTRSTATUS  == WFS_PTR_DEVONLINE)
 // The device is online,/powered on and operable

then actionTaken( do nothing);

}
```

The eventOccur( ) is triggered whenever an event notification comes from the hardware monitor agent. The actionTaken( ) is executed by the rule manager agent to give a corresponding directive to the hardware through the hardware monitor agent.

The rule manger may also report the event to the management station through the reporter mobile agent to inform the administrator to call operators to conduct a physical repair on the machine. The rule base can be updated from the management station to reflect changes in policy or hardware enhancements.

3.4 Software Monitoring Agent

The software monitoring agent structure is similar to its hardware monitoring counterparts. The agent interface to the operating system to monitor running processes. The processes that are crucial to the running of the machine are closely monitored and possible actions like restarting the process or reporting the failure of the process are performed. With most modern ATMs currently based on the Windows Operating system running different software, the Windows Management Interface (WMI) provides a common interface and object model to access management information about operating system, devices, applications, and services.

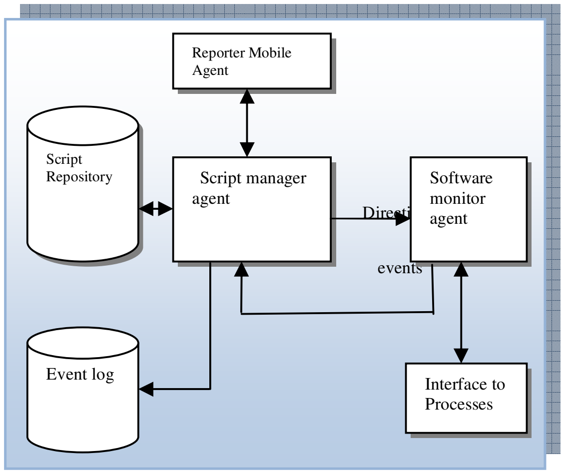

Fig 5: Software Monitor Component Architecture

Script repository contained the scripts for interrogating software state, restarts software and stops the malfunction software. The event occurrence determines which script to fire to correct the erring process.

3.5. Management Station

The management station hosts the central monitoring application. It provides the administration interface for performing various tasks such as initializing the agents, replacing the agents in the system by launching new agents, get the reports from the reporter agents, and call appropriate operators to attend to the identified problem on a particular host. It can also an update the rule repository and communicate with the agents across the networks.
The Management station is equipped with the necessary interface to facilitate seamless monitoring of all the attached ATM nodes. It is at the heart of the monitoring service and enables system administrator to have overall view of the performance of the entire network.

4. FUTURE WORK

We are working toward a prototype system to realize the architectural framework herein proposed. The details of the communication protocols and the paradigms for implementing the agents are being considered. The agent collaboration structure and the required ontology is another key focus of our future research. We hope to adopt a standardized ontology to facilitate interoperability and seamless communication among the agents

5. CONCLUSION

This work is motivated by the need of having an automated monitoring of ATM device rather than ordinary human operator. The agent-based approach we presented here is capable of performing this task with the capability of active monitoring and diagnosis of the problems. The problem resolution procedures can be updated in the rule base making the agent dynamic enough to respond to changing environments.

The benefits of this approach include:

- Reduction in mean time to repair by quickly isolating problems in critical business transactions.
- Ability to use remote diagnosis information to minimize the number of trips made to the ATM.
- Ability to monitor individual processes on the ATM and reset when necessary.
- Ability to dynamically update diagnosis rule with changing environments.

Finally agent based monitoring of ATM is a crucial solution to ensure better uptime of ATM and improve customer satisfaction, reduce operational cost and increase revenues.

REFERENCES

[1] Wu, X., Osama, D., and Phu-Dung, L (2006) "The design and implementation of a Smartphone payment system based on limited-used key generation scheme." *Proceedings of the 3rd IEEE International Conference on Information Technology*, New York, pp.62-28.

[2] Zoran, D., Ognjen, M. and Dragan, G(2007) "Internet payment system: A new payment system for internet transactions." *Journal of Universal Computer Science, 13*(4), pp. 479-503.

[3] Shintaro, K. (2003) "New perspectives on m-commerce research." *Journal of Electronic Commerce Research,6*(3) 160-164..

[4] ATM Forum Worldwide, accessed 15th August 2012. www.atmforum.com

[5] Retail Banking Research, (2008) "Global ATM market and forecast to 2013.",. http://www.rbrlondon.com/

[6] Chin S. Ou , David C. Yen , Chia-Sheng H. (2009)"Determinants of information technology investments: The case of ATM in an emerging economy" Advances in Accounting, incorporating Advances in International Accounting pp. 278–283, Elsevier.

[7] Osunade Oluwaseyitanfunmi (2010) "An Implementation of Network Configuration Management System using Mobile Agent" The Journal of Computer Science and Its Application vol.17 no 2.pp 26-35.

[8] Danny B., Mitsuru O.,(1998) "Programming and Deploying Java Mobile Agents with Aglets." Addison Wesley 1998.

[9] Guttman R., Mokas A. And Maes P.,(2009) "Agents as mediators in electronic commerce." In Intelligent Information Agents, M. Klush (Ed.), Springer Verlag, pp.131-152.

[10] Moses Ga., (1999)"Agents in E-commerce" (guest editor)*, Communications of the ACM*, Vol. 42, No.3, pp.79-80, ACM Press, March 1999.

[11] Chen K., Sycaca K, (2009.) "WebMate: A Personal Agent for Browsing and Searching"; The Robotics Institute, Carnegie Mellon University.

[12] Menczer F., Street W. and Monge A.,(2002). "Adaptive assistants for customized e-shopping. IEEE Intelligent Systems, Nov/Dec 2002, pp.12-19.

[13] K. L. Clarc, V.S. Lazarou; "A Multiagent System for Distributed Information Retrieval on the World Wide Web",

[14] D'Aloisi, D. and Giannini, V. (1995) "The Info Agent: an Interface for Supporting Users in Intelligent Retrieval." November 1995.

[15] Bohoris C., Pavlou, G., and Cruickshank, H. (2000) "Using Mobile Agents for Network Performance Management." In Network Operations and Management Symposium, pp. 637–652.

[16] Bellavista P., Corradi A., and Stefanelli, C., (2009) "An Open Secure Mobile Agent Framework for Systems Management." Journal of Network and Systems Management (JNSM) 7:3, pp.323–339.

[17] Baldi, M., Gai, S., and Picco, G. P. (1997) "Exploiting Code Mobility in Decentralized and Flexible Network Management." In Proceedings of the Workshop on Mobile Agents (MA'97) – LNCS 1219 pp. 13–26.

[18] Essien Eyo E. (2011) "Proposed Algorithm for Automated Teller Machine." In Information Technology for People- Centered Development ITePED 2011, Nigeria Computer Society Conference Proceedings Volume 22. Pp 181-186.

[19] European Committee for Standardization. Extensions for Financial Services (XFS) interface specification Release3.20 http://www.cen.eu/cen/Sectors/Sectors/ISSS/Activity/Pages/XFSCWA16374.aspx .

[20] Sohail S. (2010)," Automation of Network Management with Multidisciplinary concepts ",International Journal of Computer Technology,Vol.1(1),pp 71-77.

[21] Manoj Kumar Kona and Cheng-Zhong Xu, (**2001.**)"A Framework for Network Management using Mobile Agents" retrieved from: http://www.cic.eng.wayne.edu/paper/icec01.pdf on 03August 2012.

[22] F. Koch and C. Westphall,(2001), "Decentralized Network Management Using Distributed Artificial Intelligence", Journal of Network and Systems Management, Vol. 9(4),pp 413-424.

[23] I. El-Emary (2006),"Managing a High Speed LAN using Distributed Artificial Intelligence",Journal of Computer Science Vol.2(1), pp: 92-96.

[24] A. Alves, (2004) "Internet Traffic Engineering An Artificial Intelligence Approach", Masters Thesis, Department of Computer Science, University of Porto.

[25] Rimvydas S, Darius D and Lidija B. (2009) "Enhanced Supervision of ATM via Auto-Associative Neural Networks" , Prceedongs of Applied Stochastic Models and Data Analysis (ASMDA- 2009), Vilinius Lithunia.

[26] Cluckey S (2011) "The benefits of Real-Time ATM Management", White paper of ATM Market place, Retrieved online at http://www.esq.com/wp-content/uploads/The-Benefits-of-Real-Time-ATM-Management.pdf on 4th August 2012.