Winter 12-5-2004

Continuous Audit Agent System

L. Y. Chou Charles

Timon C. Du

Continuous Audit Agent System

Charles, L. Y. Chou, Timon, C. Du

Department of Decision Sciences and Managerial Economics, the Chinese University of Hong Kong, Hong Kong, China
{lychou, timon}@baf.msmail.cuhk.edu.hk

ABSTRACT

Recently, the provision of the real-time accounting reports over the Internet for the public corporation became popular. In response to this change on the service requirement, audit professionals considered to provide "continuous auditing" to diminish or minimize the lag time of the provision of an audit opinion behind the occurrence of business events. Though has been proposed the concept at 80's, the provided implementation models of continuous auditing are still far beyond feasible. Major obstacles to hinder wide adoption in industry lie on two issues. (1) The development of continuous auditing system should be independent to the development of client's information system. (2) The audit procedures that could be implemented by agent base automatic audit system should not be limited to the analysis on corporation internal data. In this paper, an agent base continuous auditing model and system is proposed to tackle these two major concerns. Several kinds of mobile agents are initiated by an audit agency system to simulate the behaviors of a human auditor on the audit procedures to collect various audit evidences. Each kind of mobile agent simulates a specific kind of audit procedure and acts on behalf of a human auditor to access and to inspect audit-related information hosted in the distributed information sources

Keywords: continuous auditing, multiple agent base system

1. INTRODUCTION

In recent decades, the adoption of information technology and electronic business model has dramatically changed the way we manage business and business information. Consequently, audit professionals are confronting pressing pressure to retrospect the way they have being doing auditing for decades. The pressure is coming from two directions: (1) While implementing electronic business model, the documentation for the business communication among participating parties and the information to support business operation within corporation are generated and collected in electronic forms. Under such circumstances, more and more audit evidence, which is taken by audit professionals as verification support to validate the corporation statement in their auditing process, could be collected in electronic form. (2) It is getting more corporations to disseminate their financial reports over Internet in real-time basis. The improvement in information efficiency has provided decision makers convenience in collecting useful information online. Nevertheless, such improvement turns out to be the pressure to the audit professionals to provide real time auditing service (Robert K. Elliott, 2002). In responding to this demanding request, "continuous auditing" was therefore proposed as an alternative solution. The continuous auditing here represents the assurance service where the time between the occurrence of events underlying the particular subject matter of a client and the provision of an auditor's opinion on the fairness of the disclosure of the information for the subject matter is diminished, or is minimized to an acceptable level (Alexander Kogan et al., 1999, Zabihollah Rezaee et al., 2002, Zablhollah Rezaee et al., 2001).

The research for the continuous auditing models has been on-going for two decades (Roger Debreceny et al., 2003, S. Michael Groomer and Uday S. Murthy, 1989, Harvey S. Koch, 1981, Miklos A. Vasarhelyi and Fern B. Halper, 1991). The proposals for implementing continuous auditing are executed mostly by monitoring the generation of transaction data or by analyzing the transaction data with a set of rules defined for auditing purpose. Violations to the rules are found within the transaction lifecycle or shortly after the completion of the transaction. Then certain alarms or "red flag" should be raised to call the auditor's attention in corresponding to the exceptional situation (Yining Chen, 2004). Based on the collected evidence, Auditors not only could pinpoint the weakness spot of internal controls, but also provide improving method to the internal controls of the operation systems. However, the proposals for installing facilities for continuous auditing limit their actual implementation. Commonly proposed methods for implementing continuous auditing could be classified in two approaches: embedded audit modules (EAM), and audit data mart (ADM). Both implementations lead to certain limitations for wide adoption in industry. The limitation for EAM is at its installation. Additionally, Auditors has problems to implement EAM models independently and has difficulty to embed the modules into clients' system after system being built up. The implementation of ADM is its constraints on the abilities on collecting various types of audit evidence.

In this study, a new continuous auditing model, which uses multiple mobile agents to simulate the behaviors of a human auditor on the audit procedures to collect

various audit evidences, is proposed. Each kind of mobile agent simulates a specific kind of audit procedure and acts on behalf of a humane auditor to access and inspect audit-related information hosted in the distributed information sources. As a whole, the numerous agents behave aggregately as an audit agency system. In addition to the benefits generally expected for the intelligent agent base system (Danny B. Lange and Mitsuru Oshima, 1998), we argue that this audit agency system would overcome the limitation and the concern of EAM and audit data mart. The obvious benefits provided by proposed system are listed as below:

(1) Audit procedures can be implemented continuously– The mobile agent simulates an inspector on the audit-related information, and not embedded as a sub-routine of the information system. Therefore, the mobile agent is not activated automatically by the events on the on-going basis. Rather, the mobile agent is scheduled to execute predefined audit procedures autonomously and periodically. Through a periodical inspection on the audit-related information, the audit agency system provides timely information, e.g. daily or weekly reports, depending to actual implementation requirement for various kinds of audit evidence.

(2) The development of the audit agency system is independent to the development of the information system – As for the mobile agent is a dispatched software object to the client's information system, instead of being embedded in the information system, the audit professionals could develop the audit agent system without the intensively involving the client's legacy systems. That is, to minimize the interference to the legacy system, only the authorized secure interface to the read-only access privilege to the audit related information is needed.

(3) The software agent is flexible to access various kinds of information, and the audit agency system is easy to adapt to the change of audit environment – A dispatched mobile agent interacts with local agents to acquire information. The behavior of the local agent is instructed by the local schema the needs of the mobile agents. Note that the local agent is also a software objects in which it is programmed, generated and resided in the local system. The design of the local agent provides some advantages. For example, when the audit environment changes, e.g. the changes of audit criteria, operation system, and information system, the local agent can be easily updated individually to adapt to the new environment.

(4) The mobile agents could be bestowed with different degree of intelligent activities corresponding to the needs of implementing the audit procedures. Therefore, the ability for mobile agents in executing audit procedures is not limited to the inspection of historical record in accounting information system. Rather, they can do more advanced activities such as comparing the information consistency in various applications, carrying information to the third parties system for confirmation, consulting inventory database for most updated stock level, etc. Apparently, the capability of auditing is thus expanded by the mobile agent programs.

2. AGENT BASE CONTINUOUS AUDIT MODEL

A software agent is a software object that is able to achieve a goal by performing actions and reacting to events with prearranged activities in the dynamic environment (Timon C. Du et al., 2004, Alfonso Fuggetta et al., 1998). Typically, software agents are programmed to have motilities and certain degree of intelligence. The mobility implies that the agents are able to transport themselves to given service platforms through the network while the intelligence encapsulates the artificial intelligence. Therefore, intelligent mobile agents are able to provide sophisticated computational or behavioral model to the remote data sources while interacting with the local services or applications. In this case, the mobile agent paradigm benefits the applications like electronic commerce, secure brokering, distributed information retrieval, parallel processing, monitoring and notification (Jon Woodroof and DeWayne Searcy, 2001), telecommunication networks services, workflow applications and groupware (Danny B. Lange and Mitsuru Oshima, 1998). Successfully developed prototypes of the applications include supply chain SMART project (http://smart.npo.org), brokering service in electronic market (Timon C. Du, et al., 2004), virtual enterprise (Anuj K. Jain et al., 1999), information retrieval and internet-auction house (Tuomas Sandholm and Qianbo Huai, 2000), and distributed network management (Timon C. Du et al., 2003).In the following sections, we will successively describe the major components of the multiple audit agents system for the implementation of continuous auditing.

2.1 The agent platform

In this study, we select a platform of IBM's Tokyo laboratory (http://www.trl.ibm.com/aglets/) for developing the audit agent system. The platform has a set of abstract classes and application programming interfaces (API) for users to build their own agent platform. The agent platform provides an environment of agent servers. Among those abstract classes, Aglet, AgletProxy, and AgletContext are the three most relevant key components to the agent activities -- creation, clone, dispatch, messaging, and disposal. Elaborating the functions of these three classes can help us to get a better picture of the agent platform (Danny B. Lange and Mitsuru Oshima, 1998).

(1) Aglet: Aglets is an abstract java object proposed by IBM development team. The name is a composite word

coming from "agent" and "applet". It is programmed to implement all basic agent activities we expected for the mobile agents. A system developer can model the behavior of their own mobile agents by Extending this class definition.

(2) Proxy: AgletProxies are the representatives of the Aglets. In stead of directly accessing to the Aglets public methods, the intra-Aglets messaging and the communication between Aglet and the environment all go through the interfaces of AgletProxies. This design serves for two purposes: (a). it serves as a shield to protect the aglet from malicious access and (b). the proxy also provides location transparency for the aglets, that is, it can hide the real location of Aglets. Thus, The Aglet and its proxies (more than one copy is allowed) are separated across a network. Additionally, the exchanges of message with the Aglet's proxies are just made as the interaction with local service agents.

(3) Context: Conceptually, a context is the Aglet's workplace. Every active Aglet must be hosted by an AgletContext. The function of the AgletContext is different from a server; it is a stationary object that provides a means for maintaining and managing running Aglets in a uniform execution environment. The interface to the AgletContext is used by Aglet to get information of active Aglets and their environment factors. Basically, one computer network node can run many Aglet servers (engines), one servers can host many AgletContexts, and one AgletContext could host many Aglets.

2.2 The audit agency system

In this study, the audit agent system uses the Aglet agent platform. The system functions include taking and executing the order for audit procedures from auditors to the end of consolidating the evaluation results as the reply to the auditors. The proposed system can offload human auditors' workload on labor-intensive works, and can benefit audit professionals a reliable and cost effective system to continuous and automatic execution of audit procedures.

Fig 1 shows the operations of the audit agent system. The main component of the system is the Agency Services Server (ASS), which interprets the command from auditors and initiates necessary mobile agents to complete the assignment. Then, various kinds of mobile agents are spawned to perform operations on different information sources, according to specialties of each type of audit evidences. Examples of collecting available audit evidences are as follows:

(1) Process definition confirmation: The mobile agent compares the current process definitions with the historical records and replies with the discrepancy. Such process definitions are maintained by the process control system, such as workflow management system (WfMS) or enterprise resource processing (ERP), and are identical to the conventional operation. This audit procedure assures the consistency of the business operation and alerts the auditors about the changes of the operation if any. During implementation, the mobile agent carries the process definition to the clients' site and compares against information collected from local system (agent No 1 in figure 1)

(2) Query data and precede analytical process: The mobile agent performs analytic procedures on data to check whether the inter-relationship between documents, analytical index, and company policy are

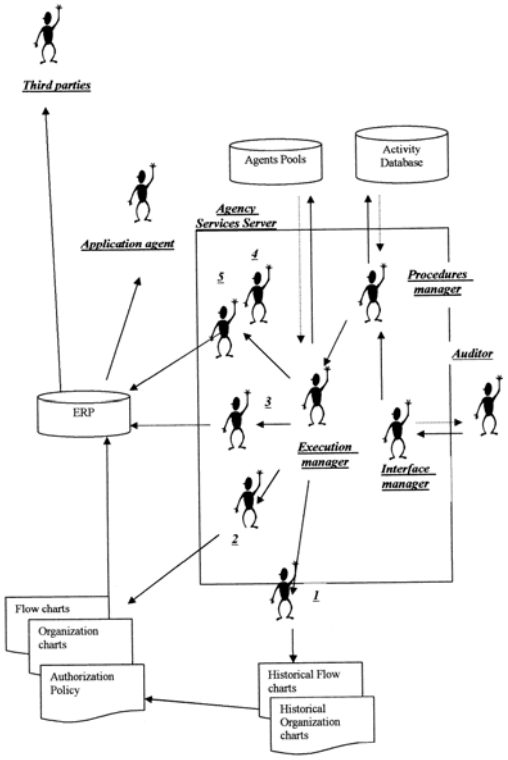

Fig 1 Audit Agency System

satisfied. By doing so, document with errors or suspicious to potential fraudulence can be allocated. During implementation, the mobile agent might visit many distributed information resources to collect related information then activate its pre-programmed analytic procedures (agent No. 2 in figure 1), or the agent can stay in ERP and complete its analytical process on the documents hosted in ERP (agent No. 3 in figure 1).

(3) Query data and take the data to third parties, e.g. customers, vendors, or bank, for further confirmation: The mobile agent retrieves and carries relevant information to the outside participants and requests their confirmation on the correctness of the information. In this design, the agent is programmed to communicate with the automatic online services of the third parties. The services can be provided by an online application or other agents communicated by standardized

communicating protocol, such as agent communication language (ACL), or Simple Agent Communication Protocol (SACP) (agent No. 4 in figure 1).

(4) Query data and take data to application agents for further confirmation: The mobile agent retrieves relevant information of the business operation in real-time and carries the information to other applications or devices for further confirmation. The example can be to examine the actual stock availability by communicating inventory management system (agent No. 5 in figure 1).

The functions of ASS include three modules: interface module, interpretation module and execution module, which are controlled by three software managers: Interface Manager, Procedures Manager, and Execution Manager (See Figure 1).

The Interface Manager manages the interfaces between the ASS and humane auditors. It is in charge of two functions: to capture required information for identification of specific audit procedure from audit requesters, and to present the execution result of the audit procedures to the audit requesters. (In adapting our study to a more automatic level, the entity for issuing audit procedures may not be limited to human auditors. An expert system which is capable to generate audit program is also technically feasible.) In order to pursue the first function, the Interface Manager needs a model to categorize the features of the audit procedures. As the conventional auditing professional has developed a framework for the organization of the comprehensive audit procedures, we will adopt the same framework. In the framework, the audit procedures are formulated as follows: audit_procedure (accounting_cycle, audit_proc_type, audit_objective, sampling_method, timing). That is, an audit procedure is specified by five procedure attributes: accounting cycle, audit procedure type, audit objectives, sampling method, and timing during the execution. The specific members (domain) of the first three properties are listed in table 1 (Alvin A. Arens, et al, 2003).

Regarding to the second function, two typical kinds of audit results will be presented: one is the error amount resulted from the execution of audit procedures, and another one is the alert to the auditors for the suspected operations that that are prone to be error or fraudulence.
The Procedures Manager is responsible for mapping the audit procedures identified by Interface Manager into a detail plan for necessary and sufficient activities to collect the audit evidence. In this plan many information are required, including the audit evidences needed to collect, the mobile agents needed to dispatch, and the configuration of the mobile agents. Moreover, the information specified in the configuration of the mobile agent includes itinerary for the agents to travel, the information sources for mobile agents to access, and the computation procedure for the specified assignment of the mobile agents. The Procedures Manager is supported by an activity database to generate suitable activities plan, where the activity database must be maintained by the auditing firm to adapt to the changes of audit requirement and client's operation system.

Properties	Members (domain)
Accounting Cycle	Sales and collection cycle Payroll and personnel cycle Acquisition and payment cycle Inventory and warehousing cycle Capital acquisition and repayment cycle Cash balance cycle
Audit procedure type	Transaction-related audit objectives Balance-related audit objectives
Audit objective	Existence Completeness Accuracy Classification Properly classified Timing (Transaction-related audit objectives) Posting and summarization (Transaction-related audit objectives) Cutoff (Balance-related audit objectives) Detail Tie-In (Balance-related audit objectives) Realizable Value (Balance-related audit objectives) Rights and Obligations (Balance-related audit objectives) Presentation and disclosure (Balance-related audit objectives)

Table 1 Domain of property value for audit procedures

The Execution Manager is responsible for initiating and administrating mobile agents to perform required assignments. The mobile agents are initiated from the Aglet's compiled java class code. Various kinds of abstract audit classes are pooled in the Agents pools and are accessed by Universal Resource Identifier from Internet. The pooling of the Aglets classes could be maintained as one of following three different ways: by local file directory (accessed by file://c:/some/path/agent.class), by Aglets server (accessed by atp://some.server/path/agent.class), or by http server (accessed by http://www.some.host/path/agent.class). The configuration of the agents is formatted as a file accessed by the agents as an input. When the mobile agents complete their assignment outside the ASS, they come back to ASS and transfer their results to Interface Manager. Then Interface Manager consolidates the audit result and formats the report to the audit requester.

3. CONCLUSION

In this study, we did not try to develop a comprehensive

system for implementing continuous auditing, which involve large efforts on the integration of several available computerized audit assist tools and a continuous audit model to seamlessly automate the whole audit process. In stead, our elaboration focuses on proposing a feasible and promising model for collection of audit evidence with continuous timeframe.

Reviewing on literature for the proposal of continuous auditing, several models have been proposed. But each proposal has its limitations in practical implementation. In fact, auditors have problems to implement EAM models independently and have difficulty to embed the modules into clients' system after system being built. Similarly, ADM has its constraints on collecting limited types of audit evidence. Fortunately, the agent-based continuous auditing model can provide auditors to have an auditing system independently on developing enterprise system. Moreover, the installation of the system can be done even the system is running and evolving.

REFERENCE:

1. Alexander Kogan, Ephraim F. Sudit, Mikios A. Vasarhelyi. "Continuous Online Auditing: A Program of Research." *Journal of Information Systems*, 1999, *13*(2), pp. 87-103.
2. Alfonso Fuggetta, Gian Pietro Picco, and Vigna Giovanni. "Understanding Code Mobility." *IEEE Transactions on Software Engineering*, 1998, *24*(5), pp. 342-61.
3. Alvin A. Arens, Randal J. Elder, Mark S. Beasley. "*Essentials of Auditing and Assurance Services: An Integrated Approach*." Upper Saddle River, N.J: Prentice Hall, 2003.
4. Anuj K. Jain, Manuel Aparico IV, and Munindar P. Singh. "Agents for Process Coherence in Virtual Enterprise." *Communication of the ACM*, 1999, *42*(3), pp. 62-69.
5. Danny B. Lange and Mitsuru Oshima. *Programming and Deploying Java Mobile Agents with Aglets*. Reading, Massachusetts: Addison-Wesley, 1998.
6. Harvey S. Koch. "Online Computer Auditing through Continuous and Intermittent Simulation." *MIS Quarterly*, 1981, *5*(1), pp. 29-41.
7. Jon Woodroof and DeWayne Searcy. "Continuous Audit Implications of Internet Technology: Triggering Agents over the Web in the Domain of Debt Covenant Compliance," *the 34th Hawaii International Conference on System Sciences*. Tennessee University. TN USA: IEEE, 2001.
8. Miklos A. Vasarhelyi and Fern B. Halper. "The Continuous Audit of Online Systems." *Auditing: A Journal of Practice & Theory*, 1991, *10*(1), pp. 110-25.
9. Robert K. Elliott. "Twenty-First Century Assurance." *Auditing: A Journal of Practice & Theory*, 2002, *21*(1), pp. 139-46.
10. Roger Debreceny, Glen L. Gray, Wai-Lum Tham, Kay-Yiong Goh, and Puay-Ling Tang. "The Development of Embedded Audit Modules to Support Continuous Monitoring in the Electronic Commerce Environment." *International Journal of Auditing*, 2003, *7*, pp. 169-85.
11. S. Michael Groomer and Uday S. Murthy. "Continuous Auditing of Database Applications: An Embedded Audit Module Approach." *Journal of Information Systems*, 1989, *3*(2), pp. 53-69.
12. Timon C. Du, Eldon Y. Li, and An-Pin Chang "Mobile Agents in Distributed Network Management." *Communications of the ACM*, 2003, *46*(7), pp. 127-32.
13. Timon C. Du, Eldon Y. Li, and Eric Wei. "Mobile Agents for a Brokering Service in the Electronic Marketplace." *available online, Decision Support Systems*, 2004.
14. Tuomas Sandholm and Qianbo Huai. "Nomad: Mobile Agent System for an Internet-Based Auction House." *IEEE Internet Computing*, 2000, *4*(2), pp. 80-86.
15. Yining Chen. "Continuous Auditing Using Strategic-Systems Approach." *Internal Auditing*, 2004, *19*(3), pp. 31-36.
16. Zabihollah Rezaee, Ahmad Sharbatoghlie, Rick Elam, Peter L. McMickle. "Continuous Auditing: Building Automated Auditing Capability." *Auditing: A Journal of Practice & Theory*, 2002, *21*(1), pp. 147-63.
17. Zablhollah Rezaee, Rick Elam, and Ahmad Sharbatoghlie. "Continuous Auditing: The Audit of the Future." *Managerial Auditing Journal*, 2001, *16*(3), pp. 150-58.