Continuous Auditing System Based on Registration Center

HUANZHUO YE, YUNING HE, ZHUOYUAN XIANG
School of Information
Zhongnan University of Economics and Law
No.114 Wuluo Road, Wuhan, Hubei, 430060
P.R.CHINA
yehuanzhuo@hotmail.com

Abstract: With the acceleration of information technologies and the availability of online real-time information systems, a rapidly growing number of organizations are conducing business and publishing business and financial reports online and in real time. Real-time financial reports provided by continuous auditing technologies are likely to necessitate continuous auditing, so what continuous auditing technologies can be utilized to facilitate continuous auditing for the next generation of accounting systems has become very important. As one of emerging information technologies, Web services technology could be seen as a good way to facilitate continuous auditing. Relying on a number of components of Web services technology, we propose a Web-service-based model using a registration center for continuous auditing called the Web-service-based continuous auditing model (WSCAM). This continuous auditing mechanism would run in the auditee's system and could be applied to provide assurance about specific business processes. In such a model, auditor could confirm specific information with the supplier of accounting materials for validation purpose, and auditee could provide specific financial information to the third party for their transaction. The frameworks and technologies which can support such a Web-service-based continuous auditing mechanism are described. The features are also presented to illustrate WSCAM.

Key-Words: Continuous auditing; Web services; XML; XBRL GL, Registration center

1 Introduction

With the development of information technology, human society has entered a new era of knowledge-based economy and globalization. Information technology has been widely used. Now accounting information systems are popularly used in many companies. More and more companies publish corporate information almost in real time, because timely financial or non-financial reports are in a high demand. As a result of this new demand, audit professionals are forced to implement auditing practices in a continuous time frame, and the time for continuous audit (CA) has come [3].

The concepts of continuous audit are now more than a decade old [16]. But the most widely accepted definition, is one released by CICA/AICPA in 1999 and reads as follows: "a methodology that enables independent auditors to provide written assurance on a subject matter using a series of auditors' reports issued simultaneously with, or a short time after, the occurrence of events underlying the subject matter"[4]. Continuous audit is likely to become commonplace for both auditor and auditee, and electronic real-time accounting systems will be used widely. Continuous audit enables independent auditors to provide some degree of assurance on continuous information in real time and detect auditor-specified exceptions from among all transactions that are processed in a real-time environment.

In this study, we propose a continuous auditing model WSCAM that uses Web services technology and has a registration center for invoking and registering services. Web services technology leverages the power of the extensible markup language (XML) and related technologies [15]. Web services have solved the interoperability of applications across operating systems, programming languages, and object models. Web services can achieve this by relying on well-supported Internet standards. Before this study, Groomer and Murthy have proposed a continuous auditing web services model (CAWS). CAWS lives in the auditor's environment, and it is a "pull" model. Based on this model, we proposed a new continuous auditing model WSCAM residing within the auditee's system, which is a "push" model.

In the following sections, we first present the background and some related researches. Section 4 introduces WSCAM, describing the specific components of the XML Web services framework that would be involved in such a mechanism. The demonstration and discussions are presented in Section 5 and 6, respectively.

2 Background

2.1 How Web Services Work

The official World Wide Web Consortium (W3C) definition follows: “A Web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols”[18].

The advent of Web services technology has changed the way computer software and the Internet, because Web services technology can solve the interoperability of applications across operating systems, programming languages, and object models [1]. Businesses and organizations can take processes and encapsulate them in a Web service and publish them on the World Wide Web. Once published, a service can be found and invoked by users around the world. Conceptually, Web services can be though of as a “stack” that embrace standards at each layer. The upper layers build upon the capabilities provided by the lower layers, and the whole web service architecture resides on the top of Internet architecture. So the simplest stack would consist of HTTP for the network layer, the SOAP protocol for the XML messaging layer, WSDL for the service description layer and UDDI for the service discovery and publication [7]. SOAP, WSDL, UDDI represent the core set of technologies to make Web services work.

2.1.1 Simple Object Access Protocol (SOAP)

SOAP is an open Internet standard used to exchange data over the Internet using Web services. It was originally proposed by IBM, Ariba, and Microsoft, and the W3C has taken on the initiative to develop it further. SOAP can let people exchange data and documents over the Internet in a well-defined way, and related standards to describe and discover Web Services.

SOAP is commonly referred to as a wiring protocol, and it is based on XML and is not a binary protocol, such as DCOM. So people can inspect the data exchange in detail using a network tunneling tool and see exactly what is going on under the hood. Although SOAP is implemented on a variety of Internet transport protocols, it most uses HTTP as its underlying protocol. So sometimes when we talk about SOAP and Web services, we may mean SOAP over HTTP [1].

2.1.2 Web Service Description Language (WSDL)

WSDL is a language using an XML format to describe how a particular Web service can be called, what arguments it takes, and so on [1].

Standardization of service descriptions to support Web services is achieved via WSDL. This language defines the interface required for interaction between a requester and a service provider and also defines the location of the service provider. A service provider publishes a service by making its WSDL description document available to potential requesters. This can be done in a variety of ways, but one standardized way is for the service provider to register the service with a registration center and for the service requester to discover the service by searching this registration center. The specification used for this registration center is the UDDI specification [7].

2.1.3 Universal Description, Discovery, and Integration (UDDI)

UDDI is a registration center where service providers publish their service descriptions and service requestors find services and obtain binding information. UDDI is implemented as a Web service itself. It is the approach for Web services.

UDDI allows businesses and individuals to publish information about themselves and the Web services they are offering. Once a Web service has been specified in WSDL, it can be made publicly available. UDDI is the preferred approach to do so. It is conceived as a global directory service, open to everybody, simple to use, and comprehensive in its scope [7].

In WSCAM, we have such a UDDI registration center residing within the auditee’s accounting system to develop continuous audit process.

2.2 XBRL GL

While Web Services can achieve data exchanging by relying on well-supported Internet standards, such as Extensible Markup Language (XML), there is no standard “XML grammar” for defining and structuring business process messages. At present, the extensible Business Reporting Language (commonly referred to as XBRL) can aim at filling this void, and we use XBRL as a standard-based reporting system being built to accommodate electronic preparation of financial reports around the world [9].

XBRL International released the core components for XBRL for General Ledger (XBRL GL). The related XBRL GL specification, which is aimed at internal accounting systems at the transaction level, is developed to provide a standardized format for representing the data fields found in transactional reports and accounting and operational systems that allows organizations to tag journal entries, accounting master files, and historical status reports in XBRL [9, 11].

"Upon generation by the client, financial statements can be tagged using XBRL, and then published to the Web. The creation of the XBRL-tagged financial statements is 'pushed' by the client system on to the Web" (J Woodroof and DW Searcy, 2001) [10]. So WSCAM is suited to using XBRL-based reporting due to the reports being produced on a "push" way. A model produced on demand (pull) is less suited to XBRL.

Besides, a XBRL GL "data hub" should be created to receive XML-based inputs from the company's internal accounting system and provide XML-based outputs to systems within or outside the company.

3 Related Research

Research on continuous auditing models has blossomed in the last two decades and explored diverse ways of achieving effective audit [2,17]. There have been several continuous auditing models, and most are merely conceptual. However, few seem to have been implemented in real time systems and the feasibility of implementing is also a problem. Generally speaking, one common approach we use for continuous auditing is embedded audit modules (EAM). Another better known model is about continuous audit model development and implementation within a debt covenant compliance domain which was proposed by Woodrocf and Searcy in 2001. In 2004, Groomer and Murthy proposed a continuous auditing web services model called CAWS, providing some important issues for our research.

3.1 Embedded Audit Modules (EAM)

Embedded Audit Modules (EAM) are subroutines within an entity's application programs that perform control or audit procedures concurrently with normal application processing. The modules are built into the client's system in the design phase as subroutines. If the modules are triggered by events, they will initiate various kinds of actions on the audit-related information. Simple tools could be utilized to capture information and reports on the violation of constraints will be generated. EAM try to supervise all of the audit-related data that resides in operational databases. They require auditors to engage intensively with the system developers in the design phase of a system, but the fact is that auditors are rarely involved in system design. So the feasibility of implementing EAM is a problem [3].

Compared to EAM, WSCAM needn't auditors to engage in the design phase of a system. Because Web services are good at integrating different web applications, an interaction with the client's system can be established after the design time and is easy to be created.

3.2 Continuous Audit Model Development and Implementation within a Debt Covenant Compliance Domain

Woodrocf and Searcy's model presents a conceptual model of continuous auditing discussed in relation to debt covenant compliance. The components of the continuous audit model are the following: the various interconnected Web servers, the continuous audit environment, the continuous audit agreement between the parties involved, the characteristics of a reliable system, the characteristics of a secure system and the evergreen reports. This model makes use of web-enabled technologies and draws attention to the need for a reliable and secure system. The need for evergreen reports is also discussed, because evergreen reports are audited reports available whenever a user accesses a web page within the CA environment [10]. The model is based on a database of transactions on the client's system, with a web interface on the auditors system for the auditor to use [14].

But the model is limited in its scope. It is discussed only in relation to debt covenant compliance. And we use XBRL as a standard-based reporting system being built to accommodate electronic preparation of financial reports around the world [9]. However, the model is less suited to using XBRL-based reporting, because the reports are produced on demand [14]. As we have mentioned, WSCAM is suited to using XBRL-based reporting due to the reports being produced on a "push" way.

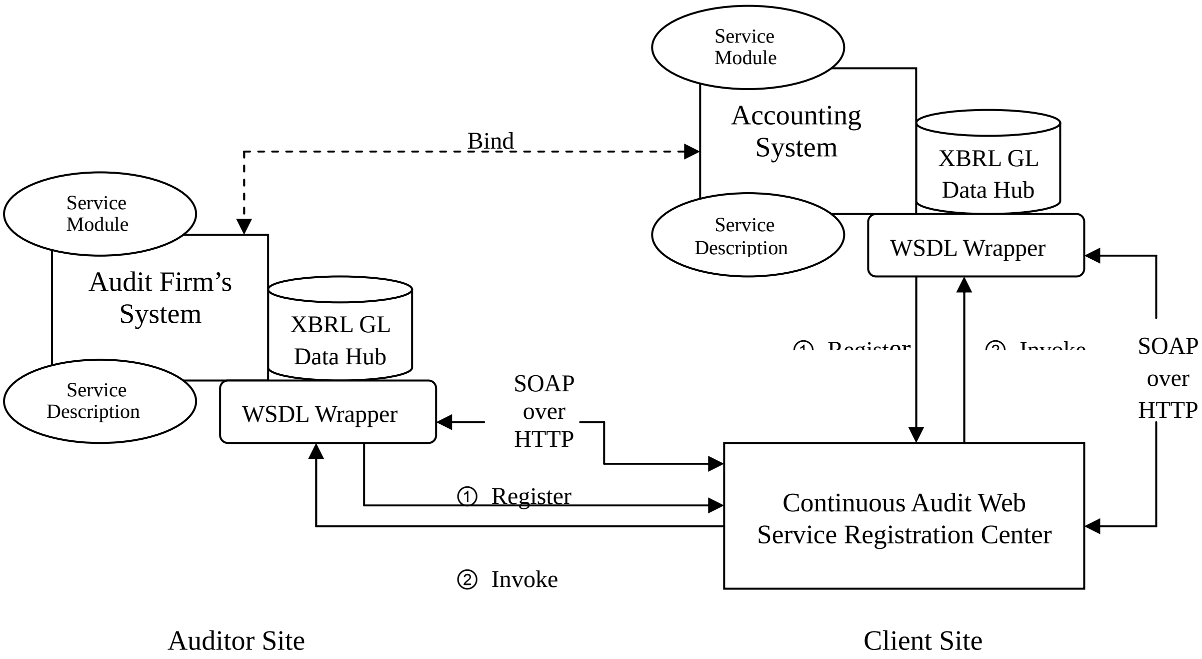

Fig.1 The interaction between auditor and auditee

3.3 Continuous Auditing Web Service (CAWS)

The paper written by Groomer and Murthy in 2004 presents a continuous auditing web services (CAWS) model that uses the emerging XML Web services framework to support a "pull" model of continuous auditing in a world of XML-enabled accounting systems. The CAWS mechanism would run as a "Web service" in the audit firm's computing environment. This approach facilitates a new "pull" model of auditing, where assurance consumers invoke the CAWS routines to obtain assurance on demand. This paper also discusses the criteria and requirements of a continuous auditing mechanism in an Internet-dominated computing environment, and describes in some detail the CAWS approach to continuous auditing and how it would operate in the context of a hypothetical sales verification system. The CAWS has two alternatives aimed at how the content of business process messages should be structured: XBRL GL and business process execution language for web services (BPEL4WS) [15].

CAWS which is a "pull" model is created on the auditor's system. But some professors point out that the CAWS approach works only if the client's systems are fully Web services enable, exposing not only the underlying data, but also the business practices. What's more, the auditor would have significant responsibility for the security of the data, potentially leading to a loss of independence at least in appearance [6]. So we propose WSCAM lodged within the auditee's computer system to solve part of the problems of CAWS.

4 The Web-service-based Continuous Auditing Model (WSCAM)

This study proposes a continuous auditing model—WSCAM, which uses Web services technology to support auditing processes. It is not lodged within the auditor's computing environment. Rather, continuous auditing functionality is defined as a set of Web services that reside within the auditee's computer system rather than the auditor's system. It is quite different from CAWS, which lives in the auditor's environment. CAWS only works when client systems will be fully Web services enabled, exposing not only the underlying data, but also the business practices. Thus, we build WSCAM in the auditee's accounting systems. It is a "push" model.

4.1 The Continuous Audit Process with WSCAM

The external auditors could use WSCAM to gather audit evidence to support an opinion on an audit client's financial statements. And the process

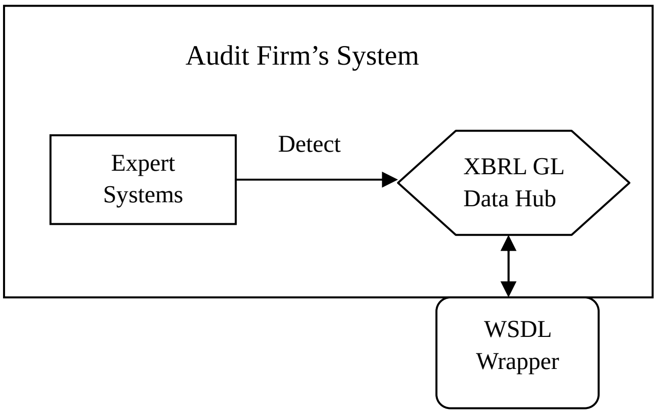

Fig.2 The fault handlers for audit exceptions

happens constantly. In WSCAM, the external auditor could publish audit services on continuous audit Web service registration center, which is quite the same as a UDDI directory, residing within the auditee's accounting system. Users of accounting system seeking these services could invoke them which are registered by auditors on continuous audit Web service registration center. Each time auditors register their services, users could invoke these services. Because after registering, auditors and users have initiated an interaction between themselves and they can bind to each other. And auditors can control the processes, because users can only use the services that are published by auditors. While users are finishing their services, these services would be provided on the continuous audit Web service registration center so that auditors can invoke them. The order of the specific process can be seen in Fig.1. Thus, the services on continuous audit Web service registration center are real time and auditing practices are implemented timely.

Fig.1 depicts the interaction between auditor and auditee, indicating how the continuous audit Web service registration center in the auditee's system interfaces with WSDL wrappers for each business process.

In terms of XBRL GL technology, the WSDL wrapper for each business process includes specific portType mappings to facilitate SOAP communication using the HTTP protocol among the auditor, auditee and the continuous audit Web service registration center. Both auditor and auditee need to convert their service description, which is prepared to publish to the registration center, to XBRL GL format. WSDL is an XML format to describe how a particular Web service can be called so that it can communicate with XBRL GL data that is also in XML format. So we need an approach to create an XBRL GL data hub for both auditor and auditee. The data hub would interface with WSDL wrapper to facilitate the continuous auditing approach proposed in this paper.

The XBRL international financial reporting standards (XBRL IFRS) is used to integrate with XBRL GL. The XBRL IFRS specification initiates for a common international XML tag set for financial reporting [12]. Users can more easily use their own data, because application service providers (ASPs) can offer to supply XBRL import and output for them [15].

Then the model must define audit exceptions. AI technologies allow every transaction to be inspected automatically, and in WSCAM we use expert systems to detect fraud and error. Expert systems are built in the auditors' systems and detecting the data invoked from the clients in the XBRL GL data hub constantly. Data is compared to the auditor-defined rules in the expert systems. Once exceptions are found, alarms may be triggered, and alerts are sent to the auditors. And then auditors can take action to notify the clients.

Fig.2 depicts how the exceptions can be found in continuous audit process.

We have mentioned that WSCAM is a model based on a registration center in the auditee's accounting systems. It is a "push" model. Thus, the continuous audit Web service registration center is created within the auditee's accounting systems. The continuous audit Web service registration center is a UDDI directory to publish business entities and the Web services they offer, and where you can find those services. It is implemented as a Web service itself and is the approach for Web services. Both auditor and auditee register and invoke their services from this registration center.

4.2 Confirming the Information with Suppliers

Sometimes, auditors need to confirm specific information with the suppliers of accounting materials for validation purposes, because business documents must be authenticated and confirmed [3]. In business transactions, confirming the information with suppliers is necessary.

Fig.3 depicts the relationship between the auditor and the supplier, indicating how the auditor interfaces with the supplier.

While the auditor is interacting with the continuous audit Web service registration center residing in the auditee's environment, an interaction

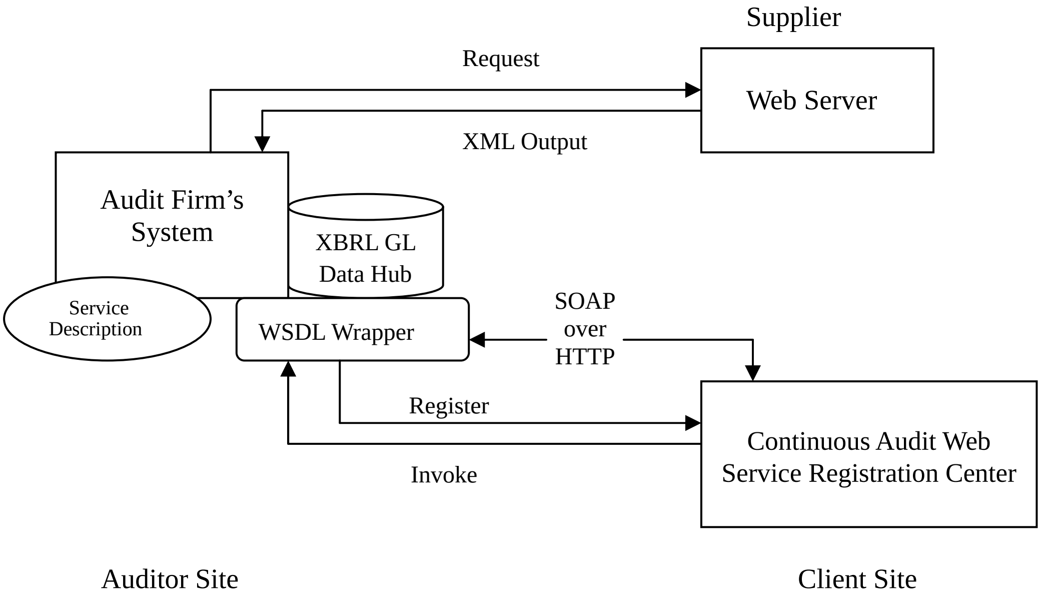

Fig.3 The relationship between auditor and supplier

between the auditor and the supplier also can be established. As the Web service registration center is in auditee's accounting system and the need for real time confirming is not essential, only a Web server is required in the supplier site. When the auditor needs to confirm with the supplier, auditor's system sends a request to the supplier for the queried records. The supplier sends back by Web pages that contain the queried records. The auditor's system accepts the records and compares them with its own records. The end auditor's system should be able to automatically consume the results, so the output should be returned in XML format. Indeed, XML output is convenient to interface with WSDL wrapper.

The supplier here is a party to send the validation for auditor's confirming and is quite different from the third party. The third party, including investors, bankers and analysts, is a user of accounting materials. And we will discuss the third party later.

4.3 The Process Invoked on Demand by Third-party Site

Not only can auditors use WSCAM to facilitate the audit process, but WSCAM can be invoked on demand by the third party. The third parties represent users of accounting materials or CA customers, such as investors, analysts, and financial institutions [15]. Sometimes, the third-party site needs to monitor the financial condition of the auditee, because the third party and the auditee may have a relationship, such as a lender and a debtor. Thus, the third party requires auditee to provide specific financial information for their transaction. The continuous audit Web registration center can be easily used to interface with the third party, because the continuous audit Web registration center is residing in the auditee's environment.

Fig.4 represents the specific process between the auditee and the third party.

In WSCAM, the third parties invoke services they need from the continuous audit Web service registration center in the auditee's system. Before invoking, auditee has to publish the related services on continuous audit Web service registration center. The third party also needs WSDL wrapper to interface with the continuous audit Web service registration center. The process is similar to the continuous audit process between auditor and auditee. But for the third parties, they needn't register to the continuous audit Web service registration center. The third party needn't to have a XBRL data hub, but the data getting from the auditee which is residing in the third-party site is also based on XBRL.

4.4 The Environment for Continuous Auditing

In WSCAM, both auditor and auditee need to have frequent interactions with internal and external applications and systems for audit evidence, which subsequently will adversely impact processing

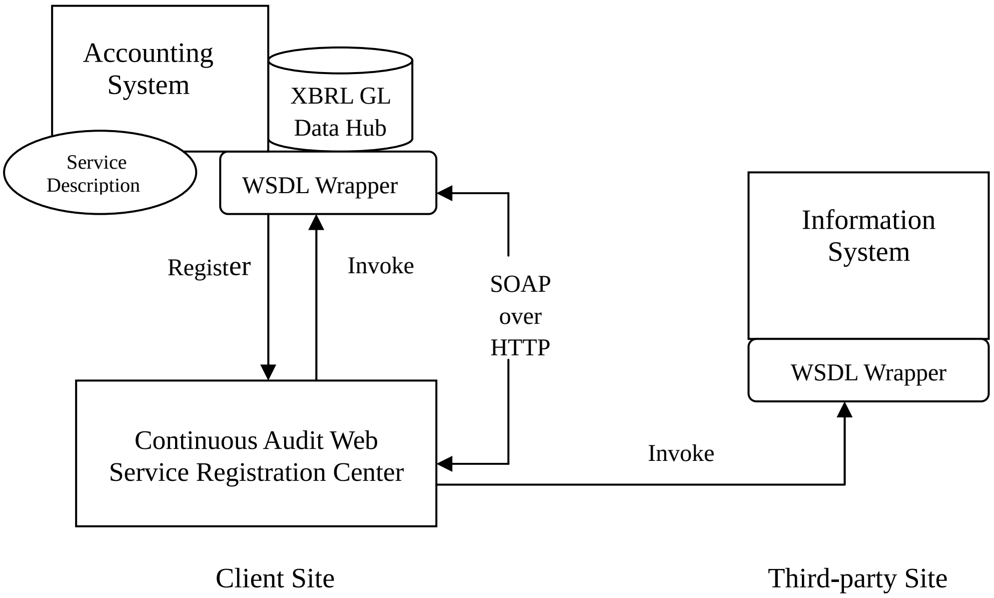

Fig.4 The specific process between auditee and third party

speed of this system. Thus, building an Intranet for auditee's system itself is essential, which obviously can speed up the auditing process and save a lot of time.

Fig.5 depicts the whole WSCAM model, indicating the relationship among auditor, auditee, supplier and third party, and showing the environment for continuous auditing.

The continuous audit Web service registration center should be built in the auditee's own Intranet so that it can communicate with the auditee's accounting system directly. And the auditee's accounting system can register or invoke the services quickly from Web service registration center which is in the same network with accounting system. The communication between the auditor and auditee is via the Internet, and the interaction between the auditor and the supplier, the auditee and the third party is also via the Internet.

To conduct a continuous audit, a number of conditions must be considered. First, the quality of both the Internet and the Intranet must be high enough so that the basic tasks, registering and invoking the services, can be finished. Second, both auditor and auditee must have highly reliable systems. Groomer and Murthy in 2004 have mentioned that: "Auditee's accounting system must be able to provide the necessary subject matter to the auditor on a timely basis. The auditor must have a high degree of proficiency in information systems, computer technology, and the audited subject matter. And finally, the subject of the audit has suitable characteristics necessary to conduct the audit. For example, if the audit is focused on evaluating internal control, then the auditor must be able to electronically interrogate these controls"[15]. Third, Woodroof and Searcy presented a model in 2001, introducing the continuous audit environment with three levels of assurance. Level 1 is an assurance regarding the reliability of the client's system and the security of transmissions of data. Level 2 is an opinion regarding the fairness of the real-time financial statements. Level 3 is an assurance on a specific analysis between the client and a third party [10].

5 Benefits and Drawbacks

5.1 Benefits

In addition to the general benefits that can be obtained with other continuous auditing models, the WSCAM model can overcome problems that are associated with the implementation of an EAM, ADM or CAWS, and assures that audit procedures to be implemented in a continuous way. The basic features and benefits are elaborated as follows.

5.1.1 The continuous auditing functionality can be realized by directly interfacing with the Web services attached to the auditee's accounting system.

CAWS is created on the auditor's system and is a "pull" model. But some professors point out that the CAWS approach works only if the client's systems are fully Web services enable, exposing not only the

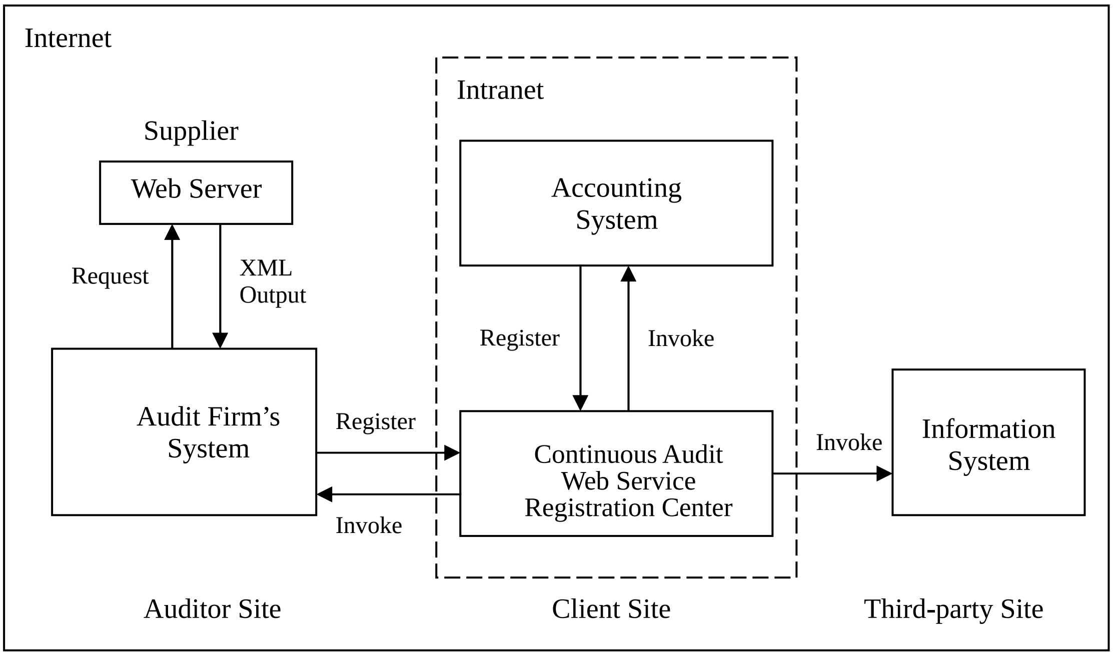

Fig.5 The WSCAM model for continuous auditing

underlying data, but also the business practices. Another weakness is that every client registers and invokes services through CAWS in auditor's system, which will definitely consume significant computing resources of auditor's system. Thus, the auditor needs a very highly reliable system that may be hard to achieve for them nowadays.

Thus, this study proposes a continuous auditing model — WSCAM lodged within the auditee's computer system rather than the auditor's system. It is a "push" model. Clients have their own continuous auditing registration centers in their systems, so they can run WSCAM with their own computing resources. That means much pressure auditors take on will be reduced. As the registration center is within every client's own accounting system, the underlying data and the business practices will not exposed to the outside.

And WSCAM requires auditee's accounting system itself to have a continuous auditing registration center. That means a continuous auditing registration center has to be installed in every auditee's accounting system, which seems to be hard to achieve in the short time. However, Web services technology is good at integration. Web services definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols. Thus, achieving the technology of creating an interface with Web services can come true soon.

5.1.2 A continuous auditing model lodging within the auditee's computer system enhances the security and solves the data ownership issues.

A continuous auditing model lodging within the auditor's system implies that all data will exposes to the outside. That means unauthorized access will be easy, because data will transport to the auditor's system and unauthorized access can get the auditee' data through entering the auditor's system first [8]. But building the model within the auditee's system means that all data still leaves in the auditee' system. The clients firstly must establish an appropriate security mechanism in their Intranet for their accounting systems. Furthermore, clients have to establish other security policy to protect their systems, such as designing digital certificates into the model.

If the model resides in the auditor's environment, the auditor would have significant responsibility for the security of the data, potentially leading to a loss of independence at least in appearance [6,14]. Thus, the model has to reside in the auditee's system. And there is no problem of the data ownership, because all data is still in the clients' system. At the same time, the model is independent of the auditor's system.

5.1.3 As Web services are good at integrating different web applications, the realization of WSCAM will be feasible.

This WSCAM model requires auditee's accounting system itself to have a continuous auditing registration center. That means a continuous auditing registration center has to be installed in every client's system and the client's system needs to be fully Web services enable.

Web service technology is a universally accepted standard. Web services perform callable functions that can be anything from a simple request to complicated business processes. Once a Web service is deployed and registered, other applications can discover and invoke the deployed service. Web services make it possible for software to access documents and run applications in a general way without requiring application specific knowledge and client software [1]. So from a technology perspective, the realization of WSCAM will be feasible.

Embedded audit modules (EAM) require auditors to engage intensively with the system developers in the design phase of a system, so the feasibility of implementing EAM is a problem [13]. Compared to EAM, WSCAM needn't auditors to engage in the design phase of a system. An interaction with the system can be established after the design time and is easy to create.

5.1.4 WSCAM is all using XML-based messages to transport.

Web Services can achieve data exchanging by relying on well-supported Internet standards, such as Extensible Markup Language (XML). XML can be seen as a standard for defining and structuring business process messages. Web services technology leverages the power of XML, and in this study we propose a continuous auditing model—WSCAM that uses web services technology. In WSCAM, we have the WSDL wrapper for each business process includes specific portType mappings. WSDL is an XML format to describe how a particular Web service can be called. And then we establish an XBRL GL data hub for both auditor and auditee, so that it can communicate with WSDL wrapper in XML format. The third party also has the WSDL wrapper, and the data getting from the auditee which is residing in the third-party site is based on XBRL. Besides, the output from the supplier is also based on XML. Thus, we have "XML grammar" for defining and structuring business process messages.

5.2 Drawbacks

Although the model offers unique benefits for continuous auditing, the WSCAM model also has some limitations in its application.

Both auditor and auditee need to have frequent interactions with the Web service registration center, which subsequently will adversely impact the performance of the system. The implementation of audit procedures consumes significant computing resources. So both auditor and auditee must have highly reliable systems.

A continuous auditing registration center has to be installed in every client's system and the client's system is fully Web services enable. It implies that corporations have to pay more money to let the client's system fully Web services enable. The cost to achieve this model will be high.

Independent auditors face new risks in providing on-demand, real-time assurance. We don't discuss much detail about new risks, but it is really a problem we have to face here. Current audit risk models are implemented for a long time, and only limited resources can be used to audit large amounts of business transactions [3]. Maybe we will discuss it in the future research and we hope others have good issues about it.

6 Summary and Conclusion

This study proposes a continuous auditing model—WSCAM that adopts Web services technology to undertake the continuous execution of audit procedures. We discuss how Web services and XML can be utilized to facilitate continuous auditing for the next generation of accounting systems. The technology of XML Web services, including SOAP, WSDL, and UDDI, is described, and XBRL is then discussed. After that, a continuous auditing model using Web services technology is proposed, which resides in the auditee's accounting system. WSCAM is designed to provide a method for continuous auditing that overcomes the problems of implementing both a continuous auditing web services (CAWS) and an embedded audit module (EAM).

This paper is a fundamental exploration of the application of Web services technology in a continuous auditing model. A next step in this line of research is to implement a prototype system demonstrating the feasibility of this model. The prototype could be developed following the XBRL GL route. Technologies such as XBRL contribute to solving the problems related to the variety of data formats which exist and the availability of audit data [14].

Future research could look at the security mechanisms of automatic audit practices and discuss the new risks we have to face in providing on-demand, real-time assurance. Based on our study, we could have more deep research, such as the specific design of the continuous audit Web service registration center, the higher degree of intelligence for continuous audit Web service registration center and expert systems, and the specific communication among auditor, auditee, supplier and third party.

Continuous audit enables independent auditors to provide some degree of assurance on continuous information in real time and detect auditor-specified exceptions from among all transactions that are processed in a real-time environment. The emergence of continuous audit will bring the society benefits, so the research of continuous auditing is valuable to our society. We have a lot work to propose a really appropriate approach and test it in order to achieve the goal of continuous auditing.

References:

[1] Adrian Turtschi, *C#.N ET Web Developer's Guide*, Syngress Publishing, Inc, 2002.

[2] Bierstaker JL, Burnaby P, Thibodeau J, The impact of information technology on the audit process: an assessment of the state of the art and implications for the future, *Managerial Auditing Journal,* Vol.16, No.3, 2001, pp.159–64.

[3] Charles Ling-yu Chou, Timon Du, Vincent S.Lai, Continuous Auditing with A Multi-agent System, *Decision Support Systems,* No.42, 2007, pp. 2274-2292.

[4] CICA/AICPA, Continuous auditing: research report, *Canadian Institute of Chartered Accountants*, 1999.

[5] DWL Searcy, JB Woodroof, Continuous Auditing: Leveraging Technology, *CPA Journal*, No.73, 2003, pp. 46–48.

[6] Eric E. Cohen, Discussion of a continuous auditing Web Services model for XML-based accounting systems, *International Journal of Accounting Information Systems*, Vol.5, No.2, July 2004, pp.165-168.

[7] Hua Zhang, Design and implementation of an XML Web service: stock information service, *McGill University*, 2003.

[8] Jagdish Pathak, A Model for Audit Engagement Planning of E-Commerce, *International Journal of Auditing*, No.7, 2003, pp.121-142.

[9] Jim Richards, XML/XBRL: IT Skills for the New Accountant, *WSCAMNZ Accounting Education Special Interest Group*, 2002.

[10] J Woodroof, DW Searcy, Continuous Audit Model Development and Implementation within A Debt Covenant Compliance Domain, *International Journal of Accounting Information Systems*, Vol.3, No.2, 2001, pp. 169-191.

[11] Neal J. Hannon, XBRL GL: The General Ledger Gets Its Groove, *Strategic Finance*, No.9, 2005, pp.57-58.

[12] Ramin, K.P., Prather, D.A., Building an XBRL IFRS taxonomy, *CPA J*, Vol.73, No.5, 2003, pp. 50–54.

[13] R.Debreceny, G.L. Gray, W.-L. Tham, K.-Y. Goh, P.-L. Tang, The Development of Embedded Audit Modules to Support Continuous Monitoring in the Electronic Commerce Environment, *International Journal of Accounting*, Vol.7, No.2, 2003, pp.169–185.

[14] S Flowerday, AW Blundell, R Von Solms, Continuous Auditing Technologies and Models: A Discussion, *Computers & Security,* No.25, 2006, pp. 325-331.

[15] US Murthy, SM Groomer, A Continuous Auditing Web Services Model for XML-based Accounting Systems, *International Journal of Accounting Information Systems,* No.5, 2004, pp. 139-63.

[16] US Murthy, SM Groomer, Continuous auditing of database applications: an embedded audit module approach, *Journal of Information Systems*, No.3, 1989, pp.53–69.

[17] Vasarhelyi MA, Concepts in continuous assurance, *Available from: http://raw.rutgers.edu/continuousauditing/conceptsincontinuousassurance13final.doc*, 2002.

[18] W3C, Web services description requirements, *W3C Working Draft* 28 October 2002.