A Collaborative Continuous Auditing Model under Service-Oriented Architecture Environments

RUEY-SHUN CHEN
Department of Information Management
China University of Technology
TAIWAN
http://192.192.85.7/Mis.htm

CHIA-MING SUN
Institute of Information Management of
National Chiao Tung University and Accounting Department of
National Yunlin University of Science & Technology
http://www.yuntech.edu.tw/~uma/
TAIWAN

Abstract: - Relying on XML and Web Service technologies under Service-Oriented Architecture (SOA) environments, this paper presents a model for continuous auditing on internal control, referred to as Collaborative Continuous Auditing Model (CCAM). By the wrapping of data transformation components with Enterprise Resource Planning databases, software venders can provide schema-matching services for the client companies of audit firms to transform data for real-time business transaction validation. Our study identifies the main components of CCAM and describes the collaborative continuous auditing processes among audit firms, client companies, schema-matching service providers, and service registries. This paper concludes with suggestions for future research.

Key-Words: - E-Activities, E-Commerce**,** Continuous Auditing, Service-oriented Architecture, Data Exchange, Schema Integration.

1 Introduction

As globalization intensifies throughout the world, the need for accurate, reliable and real-time financial information by companies, regulators, and other users has also intensified. Especially, Section 404 of Sarbanes-Oxley Act requires that each annual financial report includes an assessment of the internal control structure and procedures on the issuer that is attested by the firm's public accountants. Therefore, it is suggested that "[Auditors] will initially be used to do no more than automate existing audit procedures, and thereby take full advantage of the capabilities that it has in the new Enterprise Resource Planning (ERP) based environment" [**1**]. Modern companies utilize information technology (IT) to capture business process information at its source. This makes it possible to measure and monitor business processes at the unprecedented level of detail on the real-time basis. Thus, there is a call for "continuous auditing" or "real-time auditing" which dramatically increases the frequency of periodic audits by redesigning the auditing architecture. However, it is also argued that continuous auditing concepts seem unrealistic and highly risky to move to a model of continuous financial reporting where assurance over information is embedded in heterogeneous databases of different enterprise systems [**2**].

Some of the current challenges of continuous auditing are the obstacles to retrieving, converting and translating data from different database schema. During the last few years, there are many data exchange standards under continuous development such as Extensible Business Reporting Language (XBRL), Electronic Business using eXtensible Markup Language (ebXML), Open Financial Exchange (OFX), etc. These XML document standards can be used for data exchange among companies, financial institutions, and audit firms. However, for many companies, it is still expensive and time-consuming to translate and provide XML messages with commercial application packages, because it is complicated and laborious to search and transform data from thousands of tables in the ERP databases. Besides, it is still not clear about

how to transfer transaction documents for supporting continuous auditing or real time auditing between audit firms and their client companies.

The research approach of this study is based on the following concepts. First, the popularity of ERP systems has made the schema of ERP databases as de facto standards of data sources. In addition, the application of data exchange between companies via XML document standards is also common for many industries. Thus, we take the consideration of the continuous auditing requirements between audit firms and client companies as a problem of schema integration for heterogeneous database systems. From previous studies, schema matching is the main issue for schema integration. However, it still lacks the complete solutions for automatic schema-matching procedures and is typically performed manually. By utilizing the widely used XML document standards and existing data transformation applications developed by different companies and software venders, we can wrap these application as commercial web services that will be easy implemented under the forthcoming application environments: namely, service-oriented architecture (SOA).

One of the benefits for using SOA web service to assist continuous auditing is that each company does not have to involve great efforts on studying the format of XML documents and building application by themselves to converse data. In addition, through selecting appropriate XML transformation and security mechanisms, companies will not be afraid that others will disclose their confidential data. Under the SOA environments, the multi-agency mechanism will help the maturity and popularity of data assurance service over the Internet. By the wrapping of data transformation components with heterogeneous databases or platforms, it will create new component markets composed by many software vendors and assurance service companies to provide data assurance services for audit firms, regulators or third parties.

In brief, audit firms need to review the data integrity of business transactions with respect to the ERP database systems. Coordinating interactive activities in a continuous auditing system requires sharing data across these ERP systems. Both XML technology and web services are emergent technologies to simplify the integration and implementation efforts of data exchange between multiple database systems. Therefore, the research objective of this study is to depict a conceptual model, which combines data transformation mechanisms and web service techniques under SOA environments. The proposed model will be helpful for implementing a collaborative continuous auditing system among audit firms, client companies and third parties.

2 Background

Web-based technology extends a company's internal systems into the external environment. XML has recently emerged as a common data format for cross-platform information exchange over the Internet [**3**]. XML technology for data exchange and web services under SOA environments are the main fundamental techniques for the proposed model. These technologies and related concepts are briefly described as follows.

2.1 XML Technology and Standards for Data Exchange

The data exchange of ERP systems requires storage of XML documents. An XML-enabled database management systems (DBMS) can convert XML documents into relational representation and store them in a relational database. Thus, transmitting and integrating data in the multi-database systems can be simplified with these technologies. The main components of XML technologies for data exchange with XML-enabled DBMS are described as follows:

(1) XSLT: With Extensible Stylesheet Language Transformation (XSLT), the same XML document can be published in multiple views for different users [**4**] Therefore, XSLT can facilitate sending different subsets of the same data set to selected recipients based on particular business needs. However, writing XSLT is very time consuming and tedious. In this study, we adopt an approach to solve this problem by using an repository that stores different kinds of XSLT templates provided by many software vendors. Data transformation components can compose and decompose these XSLT templates to transform XML documents.

(2) XML schema: The concept of XML schema is very similar to the relational database schema. Thus, the schemas of relational database and XML documents can be manipulated in a similar way. One of the major schema models for describing XML document structure and for validating XML documents is the W3C XML schema. In this study, we take XML schema as a bridge for data exchange between database and XML documents.

(3) DOM: XML Document Object Model (DOM) defines an object-oriented API for XML documents. The API can be applied to access and manipulate the contained data in the XML documents. [**4**] [**5**].

For the shortage of XML-based audit tools, it is necessary to transform the XML documents into an XML-enabled database through calling DOM API. Although with these mature XML technologies, matching XML documents and ERP database schemas is still the core issue of data exchange. Schema integration is the process of constructing a global schema with a given set independently designed database schemas. In federated database approaches, each independent database has its local schema, and the local schema is translated by mapping into a common data model. Similar with the federated database approach, many industry consortia and organizations have used some pre-defined XML-based documents as industrial standards for data exchange. Employing these XML-based document standards reduces many efforts required to develop and maintain a unified schema for solving the schema integration problem among audit firms and client companies.

For audit firms, two kinds of XML document standards are used for continuous auditing: one is e-business data exchange standards, which define the data elements of business transactions, and the other is used for defining and exchanging business and financial performance information. Currently, ebXML and XBRL are the main candidates for standard XML formats in different domains. The ebXML is designed to enable a global electronic marketplace in which enterprises of any size, and in any location, could safely and securely transact business through the exchange of XML-based messages. The XBRL is used to automate business reporting, and it has occupied the unique position of being the worldwide standard for financial reporting. In continuous auditing processes, validation of transaction data with financial reports is one of the critical tasks. By adopting either ebXML or other XML e-business standards, it is easier for the software to examine the contents of the XBRL formatted financial data with XML-based transaction data.

2.2 Service-oriented Architecture

The competitive nature of globalization forces all companies to increase collaboration and cuts cost by enhancing interoperability among computer systems. A Service-Oriented Architecture (SOA) is a way of reorganizing software applications and infrastructure into a set of interacting services. These application services (web services) allow organizations to expose their system functionalities in the form of components over the Internet (or intra-net) through standard (XML-based) languages and protocols. Web services are implemented via a self-describing interface based on open standards, such Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) [**6**], and Universal Discovery Description and Integration (UDDI) [**7**].

The basic concept of SOA is a relationship of three kinds of participants: service provider, service registry, and service requestor [**7**]. The interactions among participants which is depicted as figure 1 involve the publish, find and bind operations with open protocol standards. The service provider defines the description of a web service via WSDL and publishes it to a client or a service registry. Web services are registered in a service broker with the repository of UDDI, so that potential users can find them easily. The service requestor uses a find operation to retrieve the service description from the service registry. Through SOAP over HTTP protocol, the service requestor can use the service description discovered in the service registry to bind with the service provider and interact with service implementation.

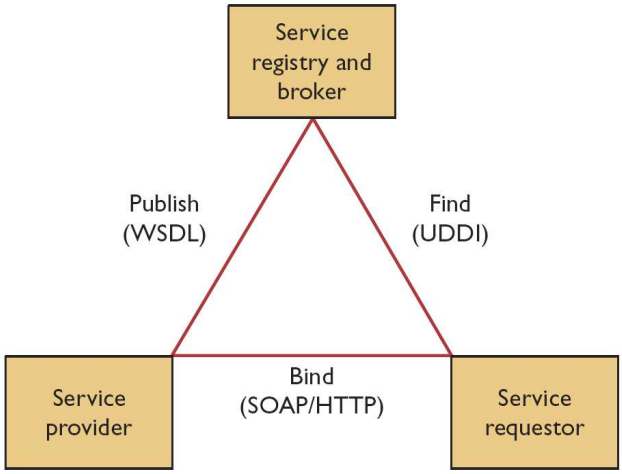

Fig. 1 Web services architecture model for SOA environments [**8**]

Unlike the traditional point-to-point architecture, SOA is comprised of loosely coupled, highly inter-operable application services. The benefit of a loosely-coupled system is its ability to accommodate changes in the structure and implementation of a service. In addition, service agility can be considered as another advantage from both the perspective of service providers and service consumers. From the view of service providers, the risk is reduced by spreading the service across a wide variety of different uses and marketplaces. From the view of service consumers, cost is reduced by the economics of scale, as well as having an efficient response to new service demands [**9**]. The

agile nature of SOA, as a result, helps organizations respond more quickly and cost-effectively to rapidly changing market conditions, technologies, policies, and regulation requirements [**10**].

3 Conceptual Model

The purpose of this paper is to present a Collaborative Continuous Auditing Model (CCAM) that employs the emerging XML web service technologies under SOA environments to support continuous auditing with client's ERP systems. The main components of CCAM are described as follows:

(1) Schema-matching repository: The repository stores the XSLT templates for various schema-matching applications between various ERP systems and the unified auditing database. The database schemas of the ERP systems are also stored in the repository as the reference for schema matching.
(2) Data transformation components: Automatic schema matching methods are still under development and domain-expert approval is required to maintain the quality of schema matching [**11**]. The complexity and time-consuming characteristic of the schema-matching problem become one of the major difficulties of executing continuous auditing activities. Thus, in CCAM, the schema matching applications, which are provided by various software vendors, are wrapped as web-service components. These components can be reused and validated by many different companies.
(3) Unified auditing database: The proposed model involves the creation of an unified auditing database, which serves as the intermediary between the internal enterprise system of the client company and the auditor's system where the continuous auditing web service resides. For the issue of data security, we use two mechanisms to protect the sensitive data of clients. First, we can remove the sensitive data elements by the XSLT template, such as customer name, product description, etc., and only keep the identifier fields and numeric data for validating the integrity of transactions. Second, we suggest keeping the unified auditing database still under the client's custody. The audit firms only have the authority to validate the data integrity of business transactions via web services, but cannot download or replicate the whole transaction data of the clients.

4 The Continuous Auditing Process of CCAM

The overview of CCAM is depicted as Fig.2 and the detailed interactive processes are depicted as Fig.3. Suppose a schema-matching service provider (SMSP) provides a transformation service between a specific XML document and a specific ERP database. Using a provider agent via a UDDI interface, The SMSP publishes a SOAP-encoded service description to a service registry (process 1). Once the schema-matching service is published, it will be used by the request agent located in the client companies. The service semantics of schema matching is defined by a WSDL file that is located in the service registry.

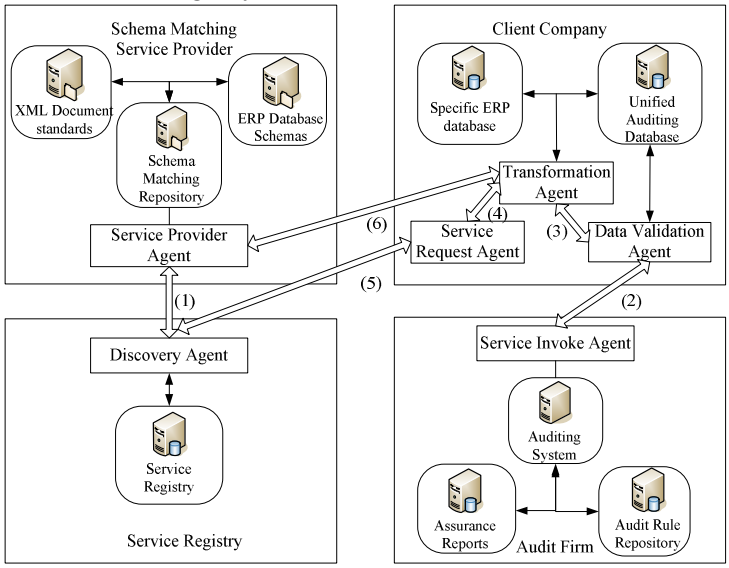

Fig 2. The Collaborative Continuous Auditing Model (CCAM)

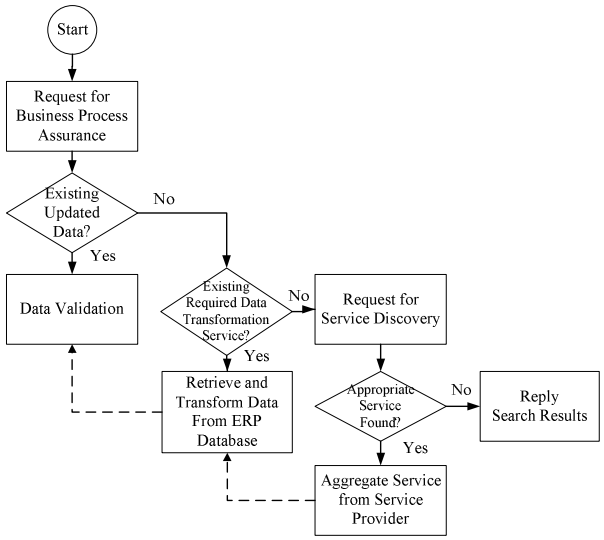

Fig 3. The Interactive Processes for Collaborative Continuous Auditing

When the audit firm has a request for internal control validation on the client's certain business transactions, the auditing system initiates an agent to invoke the data validation agent located in the client company. If it exists updated data in the unified auditing database, the data validation agent will verify the transaction data and return with the exception data that violates the audit rules (process 2).

If the tables of the unified auditing database related with the certain auditing purposes are out-of-date or empty, the data validation agent will invoke a transformation agent to retrieve and transform data from the client's ERP database (process 3). If the service of transformation agent cannot satisfy the request of data validation agent, the transformation agent will initiate a service request agent, which communicates with a service registry via the UDDI interface and transmits the service request to find the appropriate data transformation service (process 4).

When the discovery agent at the service registry receives the request from the service requester at the client company, it searches the service repository for an appropriate schema matching service. Then, the discovery agent retrieves both the appropriate service description and service semantics from service registry, and sends them to the service request agent (process 5). Using the information obtained from service registry, the requester agent in the client company sends a SOAP-encoded request message to SMSP to directly bind the service with transformation agent and invoke it. Then, SMSP provides the client company with the schema-matching component based on the request message (process 6).

Due to the numerous kinds of business transactions, the client might be unsatisfied to retrieve only one web service to complete data transformation tasks. The transformation agent can aggregate many services from different SMSP into a newly composed component.

The purpose of process 2 is to provide the auditor information referencing a particular business process. Usually, there are various audit objectives that should be achieved. Thus, when the information for auditing is not loaded into the auditing database, the data validation agent can invoke the transformation agent to transform data from the ERP database or communicate with the service request agent for invoking another cycle of schema-matching service request processes (process 4, 5 and 6).

5 Conclusions

The requirements for correct and timely financial information lead to the need for quality audit service from the auditor to support continuous verification and dissemination of accounting information [**12**]. Previously published studies for facilitating continuous auditing has focused on the use of Embedded Audit Modules (EAMs), Integrated Test Facility (ITF) or General Audit Software (GAS) [**13**] [**14**].

EAMs or ITF techniques are typically viewed as involving modification to the client's system. These modifications make them extremely expensive to implement and are likely to be resisted by the client for the negative impact with the client's enterprise systems. Therefore, this is why the GAS approach is widely used in audit firms for its independence with client's systems. However, the GAS approach is based on periodical auditing process model (PAPM), which is difficult to serve as real-time auditing for real-time business reporting. In addition, the GAS approach is suffered with the complexity and time-consuming characteristic for schema matching problems among heterogeneous databases. Therefore, it still lacks the research results on how to apply modern Internet techniques to facilitate the implementation of continuous auditing [**15**].

A continuous auditing solution must enable auditors to quickly access and retrieve data residing in any enterprise-computing platform, such as SAP R/3, Baan, PeopleSoft, etc[**16**]. Even some enterprise systems, such as SAP R/3, offer powerful auditing information systems and internal control evaluation tools, the EAMs approach still cannot satisfy all continuous auditing requirements of auditing firms.

The proposed model in this paper combines the power of web services under SOA environments with XML document standards for transforming data into a unified auditing database. The auditors can conduct testing and reporting without considering the complexities of data acquisition, transformation, and data loading. By facilitating the components of services with auditing rules, the auditing system could be equipped with proper auditing functions to meet the need of auditors and without the assistance of information system professionals.

The benefits of the proposed model are described as follows:

(1) CCAM use the database schemas of ERP package systems and XML exchange standards as de facto standards of data schema. They are widely used in many companies and well known by many software vendors. It is easy to find or

reuse data transformation components and to build schema-matching repository for these companies.

(2) With schema-matching repository composing by XSLT templates, it is convenient and efficiency to transfer the data of heterogeneous databases into a unified auditing database. It also reduces the level of complexity for the issue of schema matching.

(3) For the security issues of business transaction documents, the unified database that is under custody in the client companies could help for maintaining the confidentiality of sensitive information. Besides, through the collaboration mechanisms under SOA environments, the data auditing can still be executed remotely to validate the integrity of data and processes among transaction documents and financial statements.

(4) Another advantage following with SOA environments is the convenience for attaching additional services for different ERP packages or reusing other XML transformation components for different XML document standards.

The next step in line of this study is to implement a prototype system demonstrating the feasibility of the CCAM. In addition to the development of a prototype system, future research can explore other related issues such as how to define a unified auditing database that covers the ebXML, XBRL and the linkage between them; and how to define the audit rules that can be validated automatically by the auditing system. Future research can also compare the differences between the data-centric approach of continuous auditing techniques like CCAM proposed in this study and the process-centric approach based on predefined process models such as BPEL4WS (Business Process Execution Language for Web Services) [**17**].

References:

[1] M. A. Vasarhelyi, Alles, M.A. and Kogan, A., Principles of Analytic Monitoring for Continuous Assurance, *Journal of Emerging Technologies in Accounting*, Vol.1, No. 1, 2004, pp. 1-21.

[2] S. G. Sutton, Enterprise Systems and the Re-Shaping of Accounting Systems: A Call For Research, *International Journal of Accounting Information Systems*, Vol.7, No. 1, 2006, pp. 1-6.

[3] T. Bray, Paoli, J. Sperberg-McQueen, C. M., Maler, E., Yergeau, F., "Extensible Markup Language (XML) 1.0 (Third Edition): *W3C Recommendation*," World Wide Web Consortium (W3C), 2004.

[4] J. van den Hoven, Database Management System and XML: Interchange of Data, *Information Systems Management*, Vol.19, No. 1, 2002, pp. 94-96.

[5] H. M. D. Deitel, *XML. How to Program*, Prentice Hall, 2001.

[6] R. Chinnici, Gudgin, M., Moreau, J., Schlimmer, J., Weerawarana, S., "Web Services Description Language (WSDL) version 2.0: W3C Working Draft 1.0," World Wide Web Consortium (W3C), 2003.

[7] M. P. Papazoglou, Service-oriented Computing: Concepts, Characteristics and Directions, *WISE 2003. Proceedings of the Fourth International Conference on Web Information Systems Engineering*, 2003.

[8] M. P. S. M.N. Huhns, Service-oriented Computing: Key Concepts and Principles, *IEEE Internet Computing*, February, 2005, pp. 75-81.

[9] R. V. L. Wilkes, Service-oriented Architecture: Considerations for Agile Systems, *Microsoft Architects Journal*, April, 2004, pp. 1-17.

[10] K. H. K Zheng, TJ Pawlowski III, A Service-oriented Architecture in a Multi-agency Environment: A Case Study in Enterprise Dynamics, *10th IEEE International Enterprise Distributed Object Computing Conference Workshops*, 2006.

[11] E. Rahm and P. A. Bernstein, A Survey of Approaches to Automatic Schema Matching, *International Journal on Very Large Data Bases*, Vol.10, No. 4, 2001, pp. 334-350.

[12] K. Pany, ASB Update As of May 15, *The Auditor's Report*, Vol.21, No. 3, 1998, pp. 3-4.

[13] M. A. H. Vasarhelyi, Fern B., The Continuous Audit of Online Systems, *Auditing*, Vol.10, No. 1, 1991, pp. 110-125.

[14] U. S. Murthy and S. M. Groomer, A Continuous Auditing Web Services Model for XML-based Accounting Systems, *International Journal of Accounting Information Systems*, Vol.5, No. 2, 2004, pp. 139-163.

[15] A. Kogan, E. F. Sudit and M. A. Vasarhelyi, Continuous online auditing: A program of research, *Journal of Information Systems*, Vol.13, No. 2, 1999, pp. 87-103.

[16] Z. Rezaee, A. Sharbatoghlie, R. Elam and P. L. McMickle, Continuous auditing: Building Automated Auditing Capability, *Auditing*, Vol.21, No. 1, 2002, pp. 147-163.

[17] S. Weerawarana and F. Curbera, "Business Processes: Understanding BPEL4WS, Part 1," http://www106.ibm.com/developerworks/webservices/ library/wsbpelcol1.