2010 IEEE 6th World Congress on Services

An Analysis of Using State of the Art Technologies to Implement Real-time Continuous Assurance

Chien-Cheng Lin
Computer Science and Engineering
National Taiwan Ocean University
Keelung, Taiwan, R.O.C.
hammerlin@hotmail.com

Fengyi Lin
Department of Business Management
National Taipei University of Technology
Taipei, Taiwan, R.O.C.
linfengyi.tw@gmail.com

Deron Liang*
Computer Science and Information Engineering
National Central University
Taoyuan,Taiwan, R.O.C.
drliang@csie.ncu.edu.tw

***Abstract*—With the integrity of the information in financial reports being questioned and the shift towards more rapid financial reporting, the auditing profession has found that Continuous Assurance is an effective means of facilitating early detection of fraudulent financial reports. However, according to recent surveys, Continuous Assurance has not been widely applied to date. This fact motivates us to investigate if state-of-the-art IT technologies are capable of supporting Continuous Assurance. The contribution of this study is threefold. First, we develop an ISO/IEC 9126-based Continuous Assurance evaluation framework with six technical criteria. Second, based on the proposed framework, we review two (real-time) IT technologies, namely the Embedded Audit Module (EAM) and the Interceptor mechanism, and explore the feasibility of using them to implement real-time Continuous Assurance (CA). Overall, the interceptor approach outperforms the EAM approach, although neither approach satisfies all of the framework's technical criteria. Third, we find that using the interceptor mechanism in the middleware layer, rather than in other layers, improves the implementation of a real-time auditing interceptor. In light of the proposed evaluation framework, we consider the future development of a middleware interceptor technology that can be used to firmly establish a real-time Continuous Assurance framework.**

Keywords-Continuous Assurance; Middleware; Interceptor; Risk management; Frauds; ISO/IEC 9126

1. INTRODUCTION

The growing interest in Continuous Assurance reflects the increased attention being paid to how auditing methodologies can be used to minimize the risk of corporate fraud [1][2][15][23][36]. One argument for using Continuous Assurance is that it offers significantly improved transparency and efficiency over traditional methods. For example, in the cases of Enron and WorldCom, if the Continuous Assurance process had been applied, the auditors might have discovered the cash flow problems (both direct and indirect) in the companies' financial operations and prevented their subsequent collapse [23][35]. As a result of these failures, regulatory bodies have been prompted to focus on enterprise-risk management through Continuous Assurance [2]. In addition, Section 409 of the Sarbanes-Oxley Act (2002) stipulates that reporting must be done on a "rapid and current basis. [31]" To satisfy these requirements, reports must be accompanied by assurances about the integrity of the information [15].

Existing studies of Continuous Assurance can be divided into real-time and near real-time assurance [15]. Real-time assurance involves monitoring the data and the process of ongoing transactions, whereas near real-time assurance involves extracting and assessing data from various sources, e.g., Excel, dBase, or a general database. In this paper, we focus on real-time Continuous Assurance (the transaction data and process level). Hereafter, we use "CA" to denote "real-time Continuous Assurance," which has a broader scope than Continuous Auditing [18]. According to recent surveys [18][25][32], Continuous Auditing has not been widely implemented because current auditing tools and technologies are not suitable. This fact motivates us to investigate if state-of-the-art information technologies (IT) are mature enough to support the full deployment of CA.

The contribution of this paper is threefold. First, based on the ISO/IEC 9126 Quality Model [21] and the attributes of real-time Continuous Assurance, we propose an evaluation framework. The framework is comprised of six metrics (Continuous and Automatic Monitoring, Integrity, Usability, Maintainability, Portability, and Reliability) called real-time Continuous Assurance technical criteria. Second, we use the evaluation framework to examine two major technologies used to implement real-time CA systems, namely the Embedded Audit Module (EAM) [19], and the Interceptor mechanism [13]. Although we find that the interceptor approach is more suitable for CA than EAM, neither approach meets all the technical requirements. Third, in a CA system, there are four possible locations to install an auditing module: the application layer, the middleware layer, the operating system (OS) layer, and the network layer. After applying the proposed CA evaluation framework, we found that the middleware layer is the best location for an auditing module.

The remainder of the paper is organized as follows. The next section provides an overview of Continuous Assurance and the key challenges faced today. Based on ISO/IEC 9126, we also consider the technical criteria of Continuous Assurance in detail. Section 3 describes the proposed CA evaluation framework used to evaluate the EAM approach and the Interceptor mechanism. Moreover, to demonstrate the efficacy of Continuous Assurance as an auditing tool and encourage its use, we show how a reasonable and practical implementation of the interceptor approach on four layers could have detected a case of major fraud. In the last section,

* Corresponding author

978-0-7695-4129-7/10 $26.00 © 2010 IEEE
DOI 10.1109/SERVICES.2010.29

we summarize our findings, discuss the implications of our research, and suggest possible avenues for future research.

2. The CA evaluation framework

2.1 Why Continuous Assurance is not widely applied

Alles et al. [1] and Warren and Smith [38] observed that Continuous Assurance is broader in scope than traditional Continuous Auditing. However, scholars have not defined the terms clearly, so they are often used interchangeably. Continuous Auditing is viewed as data-intensive risk management. In general, the research community, defines Continuous Auditing as timely access to and analysis of information stored in a database (near real-time) and exchanged between EDP systems (in real-time) [22][35][39]. Continuous Auditing can therefore be treated as an operational model for internal auditors to analyze data [10].

Business processes are becoming increasingly interlinked through the use of information technology and web-based applications. As a result, real-time monitoring and assurance mechanisms that focus on the transaction process have become more important than simply accessing and analyzing data [8]. In contrast to Continuous Auditing, which is data-intensive, Continuous Assurance is defined as process-intensive, risk management. It is more efficient in detecting abnormal transactions during the process than a traditional "after-the-fact" data review because it can perform analyses across corporate business processes and address risks in a timely manner [38]. A key aspect of Continuous Assurance is that the integrity of information must be assured. Information integrity is the faithfulness of the information to the condition or subject matter that it represents. We must therefore consider both data integrity and process integrity (Sarbanese-Oxley Act, 2002) [31] in order to assure the truthfulness of information produced by a company's EDP systems.

Unlike Continuous Auditing, Continuous Assurance examines the essential components of the whole assurance process; that is, it focuses on capturing, monitoring, and analysing information derived from transactions, processes, and records to ensure the reliability and integrity of the information. However, neither method has been used very often in practice. PricewaterhouseCoopers (PwC) [28] found that 81% of the 396 companies they surveyed were either using or developing a Continuous Auditing compliant information system [18]. In a separate survey by the audit software company ACL Services Ltd. and the IIA, 36% of the auditing executives of large companies reported that their organizations had implemented Continuous Auditing, while a further 38% planned to do so in the near future [25]. The survey, called New Priorities, found that 90% of respondents thought their organizations should automate testing of internal controls at the management and business-process ownership levels. However, the results of another survey published by PwC in 2006 contradicted the findings of the previous PwC survey and the ACL survey mentioned above. It concluded that only 3% of organizations that had adopted Continuous Auditing had fully automated the process, but more than half had combined automatic and manual processes [25]. The differences in how auditors define Continuous Auditing may account for the different implementation rates reported by these surveys, and explain why CA technology has matured very slowly in practice. Judging by the considerable discrepancies in the results of the surveys, we suspect the use of Continuous Assurance in many organizations may be "continuous" in name only, as it has not been applied extensively.

Based on reports in the literature, we summarize the reasons why, from the IT perspective, CA has not been widely implemented. There are four issues as follows.

- **Issue I**: Key business information extracted from EDP systems is hard to interpret and understand.

The role and functions of Continuous Assurance in the evolution toward real-time systems can be understood within the hierarchy of control and monitoring processes throughout different levels of corporate activities [36]. Rezaee et al. [30] suggested that "an integrated auditing tool should be powerful enough for the most sophisticated analytical users." Moreover, it should have the "capacity to export the results of queries easily to common spreadsheets and database systems." In practice, a multinational organization may adopt different application systems. If the systems use different OS or applications, data transformation may be a problem. Thus different data formats represent another factor that affects the usage of *Continuous Assurance*.

- **Issue II**: Implementation costs affect the usage of auditing tools.

Rezaee et al. [30] discuss factors related to Continuous Auditing that could help reduce the costs of monitoring. According to recent studies, the implementation cost (initial construction cost) is an important factor that influences a regulatory body's willingness to use the CA platform [3][32]. Portability and maintainability are features of software that allow a developer to reuse an existing code instead of creating a new code. Portable components can easily move software from one environment to another; and maintainable components make future modifications easier when revising software for specific purposes. These are key issues in reducing development costs.

- **Issue III**: The reliability of the system should be considered when auditors apply an auditing tool.

Boritz et al. [7] suggested that business partners would have more confidence in an information system if they were provided with a SysTrust report on its reliability. To increase users' reliance on continuous reporting, an assurance mechanism must address the issue of system reliability. SysTrust and WebTrust focus on the reliability of information systems [29].

- **Issue IV**: Because of the complexity of IT technologies, *Continuous Assurance* tools may not guarantee the independence of auditing operations.

The lack of independent auditing of financial statements has perhaps dampened interest in Continuous Assurance [11]. Limiting the interaction between the auditor's CA tools and the client's system reduces the possibility of compromising

auditors' independence. The Sarbanes-Oxley Act (2002) prohibits external auditors from providing information system tools to clients they are auditing [11][39]. Even so, the auditor's tools must be able to interact smoothly and effectively with the client's system to facilitate continuous processing of data generated by the system. Thus, auditors should use independent systems that do not interfere with an auditee's operations. An independent CA system should help auditors detect and correct separation-of-duty conflicts, such as the segregation of duties and changes in key enterprise system controls, as well as define the staff's accountability to prevent unauthorized execution of transactions [6]. A configurable auditing component would allow independent auditors to change the auditing rules in real time without notifying the client's IT staff [36].

2.2 Technical criteria for real-time Continuous Assurance

We propose a CA evaluation framework for analyzing state-of-the-art IT technologies. A comprehensive CA framework should be able to fulfil the technical requirements and also address the issues discussed in Section 2.1. To determine if existing information technologies are capable of supporting such a framework, we propose an evaluation framework based on ISO/IEC 9126 [21], which is the evaluation standard recommended by the International Organization for Standardization. Specifically, the standard is used to evaluate the quality of an information system based on six quality factors, namely functionality, usability, reliability, portability, maintainability, and efficiency. We use this standard as a benchmark to assess the completeness of the technical requirements considered by the proposed CA evaluation framework.

Table 1 CA technical requirements and its mapping to ISO/IEC 9126

ISO/IEC 9126	Requirements	Description
Functionality	R1. Continuous & Automatic Monitoring	*A CA framework should support continuous and automatic monitoring.*
Functionality	R2. Integrity	*A CA framework has to ensure the integrity of information. In this paper "integrity" refers to the combined requirements of data integrity and process integrity.*
Usability	R3. Usability	
	R3.1. Understandability	*The data extracted by the CA framework should be easy to interpret and understand.*
	R3.2. Operability	*The CA framework should allow auditors to change the auditing rules without stopping the EDP system and ensure the independence of auditing operations.*
Maintainability	R4. Maintainability	*The concept of maintainability means that a module can be modified for specific tasks, can fulfil the user's modifications needs and can reduce implementation costs.*
Portability	R5. Portability	*The concept of portability means that a module can be reused, and thereby reduce implementation costs.*
Reliability	R6. Reliability	*The CA framework should not affect the normal operations of the original EDP system.*
Efficiency	--	*Since we view efficiency as an implementation issue, we do not discuss in this paper.*

Based on ISO/IEC 9126, we propose a CA evaluation framework that defines six technical requirements (shown in Table 1). In the table, Continuous and Automatic Monitoring (R1) and Integrity (R2), which are fundamental functional requirements, represent the functionality of ISO/IEC 9126. **Issue I** and **Issue IV** are two of the four issues covered by Usability (R3), which is one factor in ISO/IEC 9126. **Issue I**, which is the requirement for structured and readable information, maps to Understandability (R3.1). To facilitate independent operations, the information system should easy to use. We therefore define Operability (R3.2) to address **Issue IV**. Maintainability (R4) and Portability (R5), are quality factors that are considered by ISO/IEC 9126 and our CA evaluation framework. They are associated with reducing implementation costs (**Issue II**) as well as the operating costs after implementation. **Issue III** stresses the importance of system reliability when auditors apply an auditing tool to the auditee's information systems. To address this issue, we define Reliability (R6), which corresponds to the reliability factor in ISO/IEC 9126. In ISO/IEC 9126, efficiency refers to the performance of the software and the amount of resources used. It is affected by the complexity of the designed auditing rules; for example, an auditing component executed with one rule is usually more efficient than one executed with ten rules. Since we view efficiency as an implementation issue, we do not discuss it in this paper. Next, we discuss the six technical requirements of the proposed CA evaluation framework in detail.

- **Continuous and Automatic Monitoring (R1)**: A CA framework should support continuous and automatic monitoring.

In the *Continuous Assurance* process, data flowing through the system is continuously monitored and analysed based on a set of auditor-defined rules. Exceptions to these rules will trigger alarms that alert the auditor to any deterioration or anomalies in the system [20][24][35][39]. As auditing technology advances, the need for greater auditing efficiency and the increasing demand for real-time assurance, including regulatory compliance, are converging to drive the development of a CA framework that can support continuous and automatic monitoring.

- **Integrity (R2)**: A CA framework has to ensure the integrity of information. In this paper "integrity" refers to the combined requirements of data integrity and process integrity.

The Sarbanes Act requires both data integrity and process integrity, so transactions must be recorded [Sarbanes SEC404] and auditors need to evaluate existing procedures and controls [Sarbanes SEC303] [31]. Therefore, to provide complete assurance, auditors must ensure the integrity of information derived from EDP systems.

- **Usability (R3)**:
  - **Understandability (R3.1)**: The data extracted by the CA framework should be easy to interpret and understand.

The importance of carefully crafted official documents that prescribe professional practices cannot be overstated. The role and functions of *Continuous Assurance* in the

evolution of real-time systems must be easy to understand within the hierarchy of control and monitoring processes throughout different levels of corporate activities [36]. Clear and well-defined content can improve both the precision and efficiency of users' thought processes [14].

- **Operability (R3.2)**: The CA framework should allow auditors to change the auditing rules without stopping the EDP system and ensure the independence of auditing operations

Control of the configuration, operation, and maintenance of a *Continuous Assurance* application by auditors helps mitigate concerns about the auditors' independence and eliminates the risk of clients manipulating the CA system to prevent the detection of fraud [3][16]. In addition, it should be possible to use a CA framework to change auditing rules in real time so that auditors can gather the required information or transaction flows without notifying the party being audited. This eliminates the need for early interaction with the client's EDP systems and the need for assistance from the client's computer software specialists [32].

- **Maintainability (R4)**: The concept of maintainability means that a module can be modified for specific tasks, can fulfil the user's modifications needs, and can reduce implementation costs.

Maintainability, which is a quality factor in ISO/IEC 9126, is relevant to the effort needed to make specific modifications. From another perspective, maintainability means that some parts of the software components can be reused. Barnes & Bollinger [5] suggested that the need for cost-effectiveness in software development may be fulfiled by making the software reusable and maintainable.

- **Portability (R5)**: The concept of portability means that a module can be reused, and thereby reduce implementation costs.

The goal of portable design is to make components that are portable and reusable in several platforms. Garen [17] suggested that development costs can be reduced by a portable system design. Besides, a portable design can also reduce porting costs.

- **Reliability (R6)**: The CA framework should not affect the normal operations of the original EDP system.

The platform should have minimal impact on the performance of the client's systems. Thus, if a *Continuous Assurance* system malfunctions, it should not affect the client's EDP systems [3].

3. An analysis of using state-of-the-art technologies to support CA

Several technologies can provide near real-time assurance and real-time assurance support for the *Continuous Assurance* framework. Near real-time assurance involves extracting and assessing data from various sources, such as Excel, dBase, and general databases. This principle is used by traditional Computer-Assisted Auditing Tools/Technologies (CAATTs) [15] and the Monitoring and Control Layer (MCL) scheme [36]. On the other hand, real-time assurance involves monitoring the data and the process of ongoing transactions. Therefore, it can identify exceptions to the process in real time and alert the auditors. As mentioned earlier, in this study, we focus on real-time *Continuous Assurance* (CA). Two approaches can be used to support CA, namely, the embedded audit module (EAM) [19] and the Interceptor module [13]. A major difference between the two approaches is that EAM is embedded in the information system; hence it operates as an integral part of the system. The Interceptor, on the other hand, wraps the information system for monitoring the input and output data, so it is independent of the system.

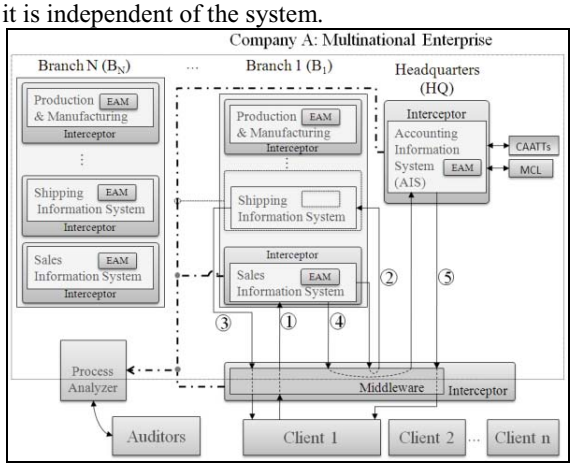

Figure 1 Illustration of the revenue cycle in a multinational organization

To determine which approach is more suitable to fulfil the CA technical requirements discussed in the Section 2, we use an example of a revenue cycle (shown in Fig. 1) to illustrate our analysis. The figure demonstrates a system architecture that connects the electronic data processing (EDP) systems of a company's headquarters (HQ), with its branch offices (B1~Bn), its clients (Client 1~Client n), and other interested parties, such as auditors and investors. The process analyzer in Fig. 1 is used to receive data from specific interceptors or EAMs to ensure the integrity of the process. In multinational enterprises, middleware, such as CORBA, SOAP framework [37] or message-oriented middleware (MOM), is usually used as a message exchange framework to transmit messages between information systems, branches, and clients. Fig. 1 shows a typical revenue cycle, which consists of the following steps: 1) the branch office B1 receives a purchase order from a client (Client 1), and processes it as an internal sales order; 2) and 3) B1 prepares the ordered products and ships them; 4) B1 sends the sales order to the accounting information system (AIS) at the HQ; and 5) an invoice is sent to the client (Client 1).

Since branches of a multinational enterprise are usually set up at different times, they may use different information systems and strategies. For example, in Fig. 1, the shipping department of B_1 did not include CA in the design phase of its shipping information system in 2000. However, all information systems used by B_n implemented CA in 2007. In 2008, the headquarters (HQ) announced that all

information systems used in the enterprise must implement CA; therefore, B_1 had to implement CA for the operational phase of its shipping information system. In the next sub-section, we evaluate the two real-time assurance approaches based on the CA evaluation framework. The analysis results are listed in the Table 2.

3.1 Applying the embedded audit module to support CA

The embedded audit model is usually incorporated with the information system in the design phase, and its functions usually reflect the control policy current at that time. In the IT domain, this is commonly called a proprietary solution. Therefore, because of its proprietary nature, EAM satisfies requirements R1 (Continuous and Automatic Monitoring) and R3.1 (Understandability).

How effective is EAM technology in helping a CA framework achieve requirement R2 (Integrity)? This requirement refers to both data integrity and process integrity, as shown in Table 1. To explain the two components, we return to the revenue cycle example in Fig. 1. We compare the purchase order (Step 1) with the corresponding sales order (Step 4). Validating the correctness and consistency of the purchase order and sales order is an example of data integrity [15]. EAM can no doubt assure the integrity of data if the control is built into the system in the design phase. Process integrity is broader in scope than data integrity. Considering the same example in Fig. 1, a CA framework must assure the truthfulness of the information from the outset in Step 1, i.e., the client site, and then through all the steps until the AIS at HQ in Step 5. In other words, all information systems involved in the process must have an auditing module to achieve this goal under the EAM approach. However, branches of a multinational enterprise are usually set up at different times and may use different information systems and strategies. Because of these factors, it is difficult to apply the EAM approach in the operational phase of information systems [9], so it is unlikely that the EAM approach can fulfil the requirement for process integrity. We therefore conclude that because the purpose of EAM is to assure the data integrity, it partially fulfils the Integrity (R2) requirement.

Issue II highlights the fact that installation costs affect the usage of auditing tools. Implementation costs and maintenance costs are the major expenditures of any system installation, such as a CA framework. Applying IT technology to ensure that an information system is maintainable and portable is an effective way to reduce such costs [5][17]. The Maintainability (requirement R4) of an EAM-based CA solution is constrained by the fact that the auditing modules must be installed in the design phase. It is very difficult to change (or maintain) them in the operational phase [9]. Furthermore, it is not easy to port an EAM to a different operating environment because the modules are designed for a proprietary system. Thus, the EAM approach does not satisfy the Portability requirement (R5).

Issue III implies that a reliable information system can increase the credibility of auditing reports; therefore, we need to consider the possible negative impact of adding auditing modules to an enterprise's information system. A component (module) failure caused by an EAM usually leads to system failure since the EAM is an integral part of the information system. Hence, it is difficult for an EAM-based CA framework to meet the Reliability (R6) requirement of CA.

Issue IV relates to the independence issue when auditors must use complex IT auditing tools. Because the EAM is an integral part of the information system, it cannot function independently. Thus, it is highly like that an auditor would need some level of assistance from IT staff when he/she operates the EAMs. As a result, it is unlikely that the EAM-based CA framework could satisfy the Operability (R3.2) requirement of CA.

The results of our analysis, which are summarized in Table 2, show that the EAM approach may not be the ideal technology to implement a CA framework since it only satisfies the six CA technical requirements partially.

3.2 Using the Interceptor approach to support CA

Since an interceptor module usually operates independently of the information system, the latter's source code is not needed for the installation and maintenance of the interceptor. Hence, the module can be installed wherever it is needed. An interceptor is applied as a wrapper that is used to wrap the information system, so it can monitor the data flowing into and out of the system. Therefore, the interceptor approach can satisfy requirement R1, Continuous and Automatic Monitoring.

In the revenue cycle example in Fig. 1, the purchase order (from Client 1 in Step 1) and the sales order (Step 4) can be intercepted by an interceptor placed in the sales information system; thus, the correctness and consistency of the intercepted data can be assured, which means the data integrity requirement (R2) can be satisfied. Recall that in the scenario discussed in Section 3.0, the shipping information system at B_1 did not implement a CA mechanism before it began operations in 2000. An auditing interceptor can be added to the shipping information system at any time because the interceptor operates as a wrapper, and it is independent of the shipping information system. Therefore, we do not need the source code or the design details of the shipping information system to install an interceptor. After the interceptor is installed in B_1's shipping information system, all interceptors are asked to send data related to the revenue cycle to the process analyzer for further analysis. As a result, the process integrity of the revenue cycle can be assured in real time if the interceptor approach is used to construct the CA framework, which means that the Integrity requirement (R2) of CA is satisfied.

Making auditing interceptors maintainable and portable is an effective way to reduce the implementation costs of CA-based systems. This also addresses **Issue II**, which we

discussed previously. Because an interceptor is an independent entity, it can be implemented and maintained in any phase of the software life cycle. Therefore, the interceptor approach fulfils the Maintainability requirement (R4). Ensuring that auditing tools are portable is another way to reduce the cost of a CA implementation. Note that an interceptor can be installed in any of the four layers at each site, i.e., the application layer, the middleware layer, the operating system (OS) layer, or the network layer. As shown in Fig. 2, an interceptor located in any one of the four layers can capture any messages flowing into or out of the information system. The messages can then be analysed if they are of interest. To facilitate our discussion later in the section, we use Fig. 2 to explain Step 4 of the revenue cycle example (shown in Fig. 1) in more detail. In this step, the sales order (SO) travels from the sales information system at B1 to the accounting information system (AIS) at HQ. Therefore, the sales order can be intercepted at four points (layers) on each site. The interceptor in each layer has unique characteristics.

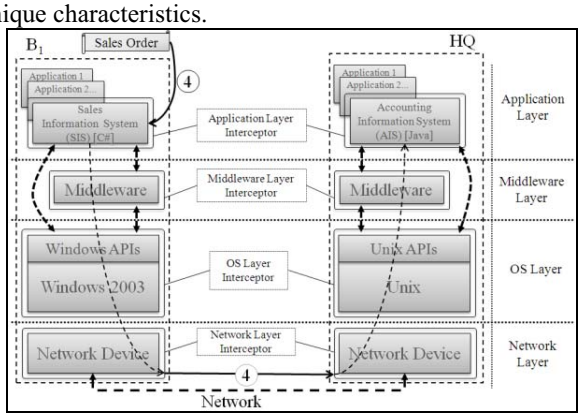

Figure 2 The four layers of an information system

To explain the differences between the interceptors in the four layers, we assume that B_1 and HQ use different operating systems (e.g., Windows 2003 and Unix respectively) and network devices. Therefore, the applications and interceptors operated at B_1 and HQ have to be compatible with their operating systems. For example, the sales information system developed by C# is used on Windows 2003 and the accounting information system developed by Java is used on Unix. Besides, to intercept messages in the OS layer, interceptors have to use different APIs; Windows APIs are used at B_1 and Unix APIs are used at HQ. In the network layer, the installation of network interceptors depends on both the operating system and the network device. By using existing products and technologies to support the interceptor approach in the four system layers, the interceptors in three of the layers (application, OS, and network) are usually constrained by the type of operating system. The network layer interceptor is limited by the differences between the network devices.

In contrast, middleware interceptors are cross platform components, so they are not affected by operating systems and network devices. Currently, there are three major middleware products: the Common Object Request Broker Architecture (CORBA) [26], the Simple Object Access Protocol (SOAP) [37], and message-oriented middleware (MOM). Only CORBA, the standard for portable interceptors [27], meets the CA Portability requirement (R5). Existing SOAP products only provide a proprietary solution. The APIs of message-oriented middleware (MOM) products, which are based on the Java Message Service (JMS) API [34], don't support the interceptor mechanism. We therefore conclude that the CA Portability requirement (R5) on the four layers is constrained, but the interceptor on the middleware layer is better than those on the other layers.

Issue I raises that the understandability of the data extracted by auditing tools is important in any CA framework. As shown in Fig. 1, before a sales order reaches the AIS at HQ, it passes through four layers of the information system, and each layer has an audit interceptor. We intercept the sales order at the application layer, the middleware layer, the OS layer, and the network layer respectively. We found the messages intercepted from the application, the OS, and the network layers are unstructured fragments of data that cannot be interpreted or understood by humans.

Windows Hook (the application layer)[1], Apache Axis handler [4] (the middleware layer)[2], Microsoft Spy++ (the OS layer)[3], and the Microsoft network monitor (the network layer)[4] are tools that can be used to implement interceptors. If other tools or technologies are used in the same layer, the intercepted result will be the same. It is clear from the figures that data intercepted in the middleware layer is structured. In contrast, data intercepted from the other three layers are fragmented, because the technologies used to implement the interceptors are usually involved with low-level system calls. We therefore conclude that only the middleware layer interceptor can satisfy the Understandability (R3.1) requirement of CA.

Because the interceptor operates as an independent entity, in theory, an auditor can modify an audit interceptor without interrupting the operations of the information system if an auditing rule is changed. However, in practice, existing interceptor technologies are not easy to modify. Currently interceptors implemented at layers other than the middleware layer require assistance from IT staff when a new version of the audit interceptor is installed. For example, if Windows Hook is used to modify the application layer interceptor, the IT staff must reinstall the interceptor and reboot the operating system. Therefore, auditors cannot

[1] API-SPY, Binary Rewriting, and Detours are widely used to support the interceptor mechanism in the application layer.
[2] Orbix 3.3 and Apache Axis [4] are used to support the interceptor mechanism in the middleware layer.
[3] System call and C library routines are widely used to support the interceptor mechanism in the OS layer.
[4] Tools like dSniff [33] and libpcap are widely used to support the interceptor mechanism in the network layer.

perform their tasks independently, as discussed in **Issue IV**; thus, the Operability (R3.2) requirement cannot be met. With regard to interceptors in the middleware layer, existing middleware products, such as CORBA and SOAP framework, provide their own proprietary solutions to address **Issue IV**. For example, the portable interceptor standard provides administration interfaces for CORBA compliant products. SOAP does not propose a standard for the interceptor; however, several SOAP products, such as Apache Axis, provide proprietary solutions with administration interfaces for the interceptor mechanism. Current message-oriented middleware (MOM) products don't provide interceptor mechanisms. In summary, some middleware technologies, such as CORBA, fully support the portable interceptor standard; therefore, they satisfy the Operability (R3.2) requirement. Others satisfy this requirement partially, and some fail to satisfy it at all.

Interceptors on the application, OS, and network layers usually use a low-level programming language, such as C language, to interact with the system's APIs. However, auditing tools developed in C language cannot prevent fatal errors, such as divided by zero. Therefore, interceptors on the above three layers cannot implement auditing tools without affecting the operations of the information system, which means they cannot meet CA's Reliability requirement (R6). Existing middleware products allow users to develop an interceptor with a compatible programming language, such as Java language, which can solve many fatal errors, such as divided by zero, which occur in C language. Therefore, a middleware layer interceptor implemented with Java language can partially fulfil the CA Reliability requirement (R6). Based on our analysis, we conclude that using the interceptor approach to support CA satisfies the Continuous and Automatic Monitoring (R1), Integrity (R2) and Maintainability (R4) requirements fully, and the Portability requirement (R5) partially. In addition, the middleware layer interceptors meet the Understandability (R3.1) requirement fully, and the Operability (R3.2) and Reliability (R6) requirements partially. The results are summarized in Table 2.

In Table 2, the tag "Y" means that the technology satisfies the CA technical requirement; "L" means the technology partially meets the CA technical requirement; and "N" means the technology fails to fulfil the CA technical requirement. Our analysis demonstrates that neither EAM nor the Interceptor approach meets all six CA technical requirements. However, overall, the interceptor approach is more effective than the EAM approach. We found that the middleware layer is the most appropriate layer for implementing the interceptor approach. Although the state-of-the-art technologies (EAM and Interceptor) cannot fulfil all the CA technical requirements, the middleware interceptor can satisfy most of them. We believe that CA would be widely applied if the middleware interceptor technology evolves to the point where it fulfils the technical support required by Operability (R3.2), Portability (R5), and Reliability (R6) under the CA evaluation framework.

Table 2 The compliance of each technology and CA technical requirements

	EAM	Interceptor			
		Application Layer	Middleware Layer	OS Layer	Network Layer
R1. Continuous & Automatic Monitoring	Y	Y	Y	Y	Y
R2. Integrity	L[5]	Y	Y	Y	Y
R3. Usability					
R3.1. Understandability	Y	N	Y	N	N
R3.2. Operability	N	N	L[6]	N	N
R4. Maintainability	L[7]	Y	Y	Y	Y
R5. Portability	N	L[8]	L[9]	L[8]	L[8]
R6. Reliability	N	N	L[10]	N	N

"Y" means the technology satisfies the requirement.
"L" means the technology meets the requirement with limitation.
"N" means the technology fails to fulfil the requirement.

4. Conclusion

Recent audit failures and corporate scandals in the United States have intensified the focus on the Continuous Assurance process as a viable risk-management tool for enterprises. With the post-Enron support for Continuous Assurance by the SEC, the AICPA and the U.S. Congress, interest in Continuous Assurance has finally reached critical mass. Several years of academic research and conferences culminated in the simultaneous establishment of centres for continuous audit research in the United States and the European Union after 2002 [2][12][30][36].

By reviewing previous studies, we show that Continuous Assurance is an essential application for supporting auditors in their work. However, our analysis indicates that Continuous Assurance has not been widely implemented. In this paper, we define the functional requirements of Continuous Assurance precisely, and examine the reasons why CA has not been widely applied by analysing two surveys and several works in the literature. Based on the functional requirements of CA and the reasons why CA has not been applied extensively, we propose a CA evaluation framework with six CA technical criteria based on ISO/IEC 9126. We then use the evaluation framework and an example of a revenue cycle to examine the maturity of two IT technologies, EAM and the Interceptor approach. Overall, the interceptor approach is more effective than the EAM

[5] The original purpose of EAM was to ensure the integrity of data.
[6] Most MOM technologies do not support the interceptor mechanism.
[7] EAM should be included in the design stage, as it is difficult to implement in the operational phase.
[8] Portability of the application, OS, and network layer interceptors is constrained by OS and network device.
[9] Proprietary solutions are only portable by their respective products (Axis and XFire handler, and MS-WSE filter).
[10] Interceptors developed by C or C++ language may cause fatal errors (e.g., divided by zero) if they are not carefully developed.

approach; and the middleware layer is the most appropriate layer for implementing the interceptor approach. Although the EAM and Interceptor approaches cannot fulfil all of CA's technical requirements, the middleware interceptor can satisfy most of the requirements.

To improve the applicability of CA, we believe there should be further investigation of the Operability (R3.2), Portability (R5), and Reliability (R6) requirements of middleware technologies to support CA, including new research on interceptor technology. Other research directions include developing user-friendly mechanisms and tools for auditors who do not have a strong IT background, and more intelligent tools to assist auditors with better auditing rules and policies.

References

[1] M.G. Alles, A. Kogan, and M.A. Vasarhelyi, "Feasibility and economics of continuous assurance," *Accounting: A Journal of Practice & Theory*, vol. 21, no. 1, pp. 125-138, 2002.

[2] M.G. Alles, A. Kogan, and M.A. Vasarhelyi, "Restoring auditor credibility: tertiary monitoring and logging of continuous assurance systems," *International Journal of Accounting Information Systems*, vol. 5, no. 2, pp. 183-202, 2004.

[3] M.G. Alles, G. Brennan, A. Kogan, and M.A. Vasarhelyi, "Continuous monitoring of business process controls: a pilot implementation of a continuous auditing system at Siemens," *International Journal of Accounting Information Systems*, vol. 7, pp. 137-161, 2006.

[4] Apache, Apache Axis architecture guide, Apache Axis 1.x Documents, 2006, Available at: http://ws.apache.org/axis/.

[5] B.H. Barnes, and T.B. Bollinger, "Making reuse cost-effective," *IEEE Software*, vol. 8, no. 1, pp. 13-24, 1991.

[6] P.V. Boccasam, and N. Kapoor, "Managing Separation of Duties Using Continuous Monitoring," IT Audit, 6. Altamonte Springs, FL: The Institute of Internal Auditors, Available at: http://www.theiia.org/itaudit/index.cfm?fuseaction=forum&fid=5433, 2003.

[7] E. Boritz, E. Mackler, and D. McPhie, "Reporting on Systems Reliability," *Journal of Accountancy*, vol. 188, no. 5, pp.75-87, 1999.

[8] C.E. Brown, J.A. Wong, and A.A. Baldwin, "A Review and analysis of the Existing Research Streams in Continuous Auditing," *Journal of Emerging Technologies in Accounting*, vol. 4, pp. 1-28, 2007.

[9] C.L. Chou, T. Du, and V.S. Lai, "Continuous auditing with a multi-agent system," *Decision Support Systems*, vol. 42, no. 4, pp. 2274-2292, 2007.

[10] D. Coderre, "A continuous view of accounts," *The Internal Auditor*, vol. 63, no. 2, pp. 25─31, 2006.

[11] H. Du, and S. Roohani, "Meeting Challenges and Expectations of Continuous Auditing in the Context of Independent Audits of Financial Statements," *International Journal of Auditing*, vol. 11, pp. 133-146, 2007.

[12] R.K. Elliott, "Twenty-First century assurance," *Auditing: A Journal of Practice and Theory*, vol. 21, no. 1, pp. 139-146, 2002.

[13] P. Felber, and P. Narasimhan, "Experiences, strategies, and challenges in building fault-tolerant CORBA systems," *IEEE Transactions on Computers*, vol. 53, no. 5, pp. 497-511, 2004.

[14] I.E. Fisher, "On the structure of financial accounting standards to supporting digital representation storage, and retrieval," *Journal of Emerging Technologies in Accounting*, vol. 1, pp. 23-40, 2004.

[15] S. Flowerday, and R. Solms, "Real-time information integrity=system integrity+data integrity+continuous assurances," *Computers & Security*, vol. 24, pp. 604-613, 2005.

[16] S. Flowerday, A.W. Blundell, and R.V. Solms, "Continuous auditing technologies and models: a discussion," *Computers & Security*, vol. 25, pp. 325-331, 2006.

[17] K. Garen, "Software Portability: Weighing Options, Making Choices," *The CPA Journal*, vol. 77, no. 11, pp. 3, 2007.

[18] M. Green, "Business look to continuous auditing, monitoring," *Best's Review*, vol. 107, no. 4, pp. 76, August 2006.

[19] S.M. Groomer, and U.S. Murthy, "Continuous auditing of database applications: an embedded audit module approach," *Journal of Information Systems*, vol. 3, no. 2, pp. 53-69, 1989.

[20] ISACA Standards Board, "Continuous monitoring: is it fantasy or reality?" *Information Systems Control Journal*, vol. 5, pp. 43-46, 2002.

[21] ISO-9126, Software engineering -- Product quality, 2001, Available at: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22749.

[22] A. Kogan, E.F. Suit, and M.A. Vasarhelyi, "Continuous online auditing: a program of research," *Journal of Information System*, vol. 13, no. 2, pp. 87-103, 1999.

[23] J.R. Kuhn, and S.G. Sutton, "Learning from WorldCom: implications for fraud detection through continuous assurance," *Journal of Emerging Technologies in Accounting*, vol. 3, no. 1, pp. 61-80, 2006.

[24] D. Liang, F. Lin, and S. Wu, "Electronically auditing EDP systems─with the support of emerging information technologies," *International Journal of Accounting Information Systems*, vol. 2, pp. 130-147, 2001.

[25] T. McCollum, "Continuous auditing on the rise," *The Internal Auditor*, vol. 63, no. 4, pp. 15-16, 2006.

[26] Object Management Group, Common Object Request Broker Architecture: Core Specification, V 3.0.3. OMG Technical Committee Document formal/04-03- 01.

[27] Object Management Group, CORBA Portable Interceptors, Common Object Request Broker Architecture (CORBA) v3.0.3, http://www.omg.org/cgi-bin/doc?formal/ 04-03-19.

[28] PwC, PricewaterhouseCoopers 2006 State of the Internal Audit Profession: Continuous Auditing Gains Momentum.

[29] A. Pugliese, and R. Halse, "SysTrust and WebTrust: Technology assurance opportunities," *The CPA Journal*, vol. 70, no. 11, pp. 28-32, 2000.

[30] Z. Rezaee, A. Sharbatoghlie, R. Elam, and P.L. McMickle, "Continuous auditing: building automated auditing capability," *Auditing: A Journal of Practice & Theory*, vol. 21, no. 1, pp. 147-163, March 2002.

[31] Sarbanese-Oxley Act. (2002), United States of America 107th congress. US Congress. Available at: http://www.sec.gov/ bout/laws/soa2002.pdf.

[32] D.L. Searcy, J.B. Woodroof, and B. Behn, "Continuous audit: the motivations, benefits, problems, and challenges identified by partners of a big 4 accounting firm," Proceedings of the 36th Hawaii International Conference on System Sciences, 2003, pp. 1-10.

[33] D. Song, dSniff document (dSniff 2.3), Available at: http://www.monkey.org/~dugsong/dsniff/, 2001.

[34] Sun, Java Message Service (JMS), Sun Microsystems Inc., 2010, Available at: http://java.sun.com/products/jms/.

[35] M.A. Vasarhelyi, "Would continuous audit have stopped the Enron mess?" Working paper, Rutgers University. 2005.

[36] M.A. Vasarhelyi, M.G. Alles, and A. Kogan, "Principles of analytic monitoring for continuous assurance," *Journal of Emerging Technologies in Accounting*, vol. 1, pp. 1-21, 2004.

[37] W3C, SOAP Version 1.2 Part 0: Primer, W3C Recommendation, 2007, Available: http://www.w3.org/TR/soap12-part0/.

[38] D. Warren, and M. Smith, "Continuous auditing: an effective tool for internal auditors," Internal Auditing, vol. 21, no. 2, pp. 27-35, March/April 2006.

[39] J. Woodroof, and D.L. Searcy, "Continuous audit model development and implementation within a debt covenant compliance domain," International Journal of Accounting Information Systems, vol. 2, pp. 169-191, 2001.