Continuous auditing for web-released financial information

Chi-Chun Chou
Department of Accounting Information, National Taipei College of Business, Taipei, Taiwan, Republic of China, and
C. Janie Chang
Charles W. Lamden School of Accountancy, College of Business Administration, San Diego State University, San Diego, California, USA

Abstract

Purpose – The increasing provision of timely financial information through web-based technology is expected to improve the quality of communication between a company and its stakeholders. However, the information asymmetry problem still exists since almost all "web-releases" usually remain unaudited. The purpose of this paper is to propose conceptual and technical frameworks of continuous auditing to provide a solution for this problem. This solution could also move the traditional auditing forward to the new e-auditing generation.
Design/methodology/approach – This paper develops a conceptual framework to present why continuous auditing would dominate other auditing approaches in examining web-based financial information. Using a 3 × 2 × 2 × 1 design, this study compares the economic efficiency of three auditing approaches under the joint-combination of various disclosure types, materiality perceptions and information environments. A technical framework, the external continuous auditing machine, is derived from the conceptual framework to specify the generic procedures to perform the online control testing and the continuous substantive testing over web-releases.
Findings – Continuous auditing issues are scrutinized both theoretically and technically. Two main conclusions arise. First, the behavior model simulates various information disclosing and auditing environment and argues that the continuous auditing would be the most appropriate approach for web-releasing assurance. Although the hypothesis derived from that model still needs further empirical supports, the anticipated sustaining is quite reasonable under the emergent web-release practice.
Originality/value – Given the new era of online, real-time business reporting, constructing a theoretical model and applying it to develop a technical model for implementing continuous audits for web-releases provide significant contributions to the accounting/auditing professionals as well as researchers.

Keywords Auditing, Disclosure, Internet, Tests and testing, Online reporting
Paper type Research paper

1. Introduction

Timeliness is one of the most important qualitative characteristics of financial information. Through years, standard-setting bodies like Accounting Principles Board (APB) (1970), Financial Accounting Standards Board (FASB) (1979), Canadian Institute of Chartered Accountants (CICA) (1972) and Institute of Chartered Accountants in England and Wales (ICAEW) (1975) consistently emphasized the importance of timeliness of financial information in their formal statements. Despite the consentaneous highlights from various parties, information users and accounting academics, the accounting professionals have long faced the problem that they could not provide adequate and sufficient timely financial information as demanded. According to some previous investigations (Ettredge *et al.*, 2006, 1994; Collins, 1994; Sinclair and Young, 1991; Zeghal, 1984; Penman, 1984; Chambers and Penman, 1984),

Review of Accounting and Finance
Vol. 9 No. 1, 2010
pp. 4-32

1475-7702
DOI 10.1108/14757701011019790

perhaps the major reason why timeliness is so difficult to pursue is that mature information technology is not available.

For a long time, due to the unavailability of both real-time processing and disseminating technology, companies can only rely on the third-party media (e.g. such as newspapers, magazines, or the database of formal reports submitted to the SEC), other than their own disclose system, to provide *untimely* information. However, the situation has been changing dramatically in the past decade. The online transaction processing technology and the innovative Internet technology definitely make the production and dissemination of real-time accounting information possible. The increasing demand from the capital market for direct-access to company real-time information, the rapid growth of e-business technology brought many public companies to disclose the selected financial and business information on their web sites voluntarily. Through their web sites, most companies exercise their autonomy and mobility in the decisions of reporting contents, timings and forms. In fact, given US Securities and Exchange Commission's disclosure requirements, most public companies would use web-releases[1] to disseminate their important financial and operating information on the internet. Although the increasing provision of using timely web-releases is expected to strengthen the efficiency of communicating financial information, behind those accounting numbers, the information asymmetry problem still exists and could harm both the reporting companies and information users, since they are usually unaudited. It is a great concern that the unaudited financial reports/information been globally disseminated without promise on their reliability.

The regulatory bodies like PCAOB, AICPA and CICA have raised concerns of the emerging auditing problem for web-release information. Many recent studies have emphasized the importance of continuous auditing, since in the future the major audit objectives will become the real-time business reporting on internet (El-Masry and Reck, 2008; Chou *et al.*, 2007; Searcy and Woodroof, 2003). Therefore, it is the crucial moment for the accounting/finance profession to explore how continuous auditing can be implemented to fulfill users' needs in getting real-time information, and why this approach can be used to solve the reliability issue in making judgments on web-based financial information.

In consideration of the deficiency of conceptual and technical foundation, the "continuous auditing report" (AICPA and CICA, 1999) has further explored and identified the requiring characteristics of continuous auditing. This report, based on the recognition of the issue in web-based financial information, also tried to re-examine and propose challenges to the existing auditing standards. Through their sophisticated study, the basic concepts of continuous auditing are clarified and becoming a formal standard in the near future. However, due to the constraints of their missionary goals, this research report did not provide much insight in a way to build up a theoretical framework for continuous auditing. For instance, what exactly is the economic definition and implication of continuous auditing? Would continuous auditing be most efficient to audit web-releasing information? Also, how to put the new continuous auditing concepts into practice? Using what kind of information technology, the approach could be successfully applied and implemented?

To answer the aforementioned questions, our research efforts are to establish both conceptual and technical foundation of continuous auditing. This study attempts to explore the following issues. First, we analyze various auditing approaches to examine, from an economic view, whether continuous auditing will dominate other approaches in auditing web-based financial information. Based on the analyses, several testable hypotheses are proposed. Second, an accounting information system framework for continuous auditing is developed to lay a well-formed technical foundation for testing the above hypotheses. In this stage, both the generic framework[2] of online control testing (OLCT) and continuous substantive testing model (CSTM) for continuous auditing are constructed by using formal system modeling tools and object-oriented technology.

2. Analysis of continuous auditing approaches

In this section, to determine whether continuous auditing is better than other auditing approaches regarding the web-releasing practice, we examine and compare the different natures of various auditing approaches in virtual capital markets where the information asymmetry problem between the information reporter and receiver exists. It is assumed that information reliability can be achieved only by using third-party, qualified auditing service. To attain disclosure efficiency (i.e. timely information), companies would like to have reliable web-releases. Hence, the objective of our model is to reduce the cost of auditing on web-releases. We conduct cost/benefit analysis among different auditing approaches under various informational scenarios. Several important variables and concepts are notated using mathematical characters or symbols in the following model.

2.1 Basic assumptions

Some unexpected factors, suggested by previous information-related literature (Verrecchia, 1983, 1990), might increase the complexity of hypothetical web-release scenarios. For example, concerning the degree of disclosure efficiency, would web-releases be expected to spread the financial information over the capital market immediately? Also, in a voluntarily web-releasing environment, some companies might have competitive advantages while releasing its important operating information on the web. However, the expected penalty from inaccurate information disclosure might also discourage the same companies' web-release decisions. After considering all undesired factors, the following three assumptions must be made to avoid possible confusions to our conceptual framework:

(1) information sequence assumption;

(2) disclosure efficiency assumption; and

(3) audit quality assumption.

The information sequence assumption is important. In the conceptual framework, all public events are assumed to be observable by all market participants right after events/information announced through some "bulletin board" mechanism. However, not all of the participants are informed with the inside information of those events. Therefore, the events would create an information sequence starting from the *event occurrence*, then the *state change*, the subsequent *information need* and the final *trading decision*. This sequence is assumed in our theoretical model.

The second assumption is about disclosure efficiency. Information holders are assumed to be willing to voluntarily disclose their private information to avoid the reinforcing capital costs from the market. However, the proprietary cost and information precision problems are assumed to cause minimal influence in our model so that all companies will have the homogeneous incentive to release financial information on their web sites.

The audit quality assumption presumes that no moral hazard or opinion shopping exists in our phenomena and auditors are endowed identically sufficient audit technology to detect and report all possible errors, frauds or irregularities.

2.2 Definitions of variables

This section provides rigorous definitions of the variables used in the conceptual framework. The definitions and the following mathematical notations are based on common accounting, finance and auditing knowledge, so they are not referred to any specific literature.

The first variable is *Event* (denoted as *e*). *Events*[3] are those public-observed economic incidents expected to affect a company's value. Events might not be recognized in a company's book based on generally accepted accounting principles (GAAP), but they are indeed related to its economic value. For example, the obtainment of a large long-term sales contract and the re-negotiation of employee pension fund plans would be recorded in books. Conversely, the resignation or retirement of an important R&D team member for an intelligence-intensive company, the significant price fluctuations of a major material for a manufacturing company or the unpredictable volatility of market rate for an investment bank cannot be included in accounting books. After the occurrence of an important event, it is reasonable to assume the uninformed market would have the desire to obtain some new accounting numbers in order to make a better estimate of this event's economic impact.

The variable *Information Request* is denoted as *q*. After the occurrence of a material event, the market will request timely private information to improve their precision of estimate for target companies, including important accounting numbers. In the case of a price-protection market, this phenomenon may also be interpreted as the company's voluntary need to mitigate the severe agency cost reinforced by the market.

The variable *Transaction* is denoted as *t*. Transactions are financial activities that can be measured, reasonably estimated and recognized by accounting standards. They might or might not be related to the company's market value, and might or might not be publicly observable. For example, financial transactions like credit sales, inventory acquisition, stock issuance, payment collection, etc. might alter both the company market value and book value while they occur. But accrual items such as interest payables or receivables, depreciation expense and other adjusting entries recorded at the end of each accounting period are paper transactions with no influence to the company's market value. Transactions are the original source where auditors can collect auditing evidence and assess a company's assertions.

The variable *Disclosure* is denoted as *d*. The private information holder can release financial information in two ways, the first way is to provide up-to-date accounting data on the web regarding each information request for an individual event. This timely disclosure *dr* will be defined as the *real-time* disclosure. On the other hand, *dp* denotes the traditional periodical disclosure of accounting data disseminated on a monthly, quarterly or yearly basis. In other words, *dr* is a timely and event-triggering disclosure, while *dp*, in contrast, is untimely and periodical.

In our framework, *Au* represents the set of various auditing approaches. *CAu* and *RAu* denote two kinds of timely auditing[4], the *continuous auditing* and the *real-time auditing*, respectively. *PAu* denotes the traditional *periodical auditing*. We will explain these approaches in the next section. As the audit quality assumption described, the basic audit quality and the capability of discovering and reporting material misstatement of financial information is assumed to be identical among the three approaches. However, there exist two major differences to distinguish them. One is the frequency and timing of testing programs conducted, the other is the frequency and timing of opinions issued by auditors.

The variable *Audit Opinion* is denoted as *AO* representing the opinion in the audit report attached along with the company's financial statements and reports. The last variable is *Decision* denoted as *D*. After each event occurs, regardless of the provision of *dr* (timely disclosure), the market traders will adjust their beliefs and make buy or sell decisions in terms of the information sequence assumption.

2.3 Conceptual framework

Based on the variables defined in the previous section, we propose a conceptual framework of continuous auditing. The purpose of the framework is to establish a system so that further investigations of the performance of each competing auditing approach could be demonstrated more clearly. The concepts of seven components formed by the predefined variables of our framework are:

(1) information request environment;

(2) information cycle;

(3) disclosure approach;

(4) disclosure efficiency;

(5) audit approach;

(6) audit efficiency; and

(7) disclosure efficiency.

Information request environment represents the types of one market's information demand: the most *timely information request environment* (TIRE) and the *periodical information request environment* (PIRE). *TIRE* assumes a set of information requests always exists after any event occurs. PIRE, conversely, is assumed there are no requests for information disclosure in terms of any occurring events. The information requests are corresponding with the accounting periods only[5]. Between the two extreme cases, there are *near timely* or *near periodical* environments as we normally observe in the real world.

The information cycle is defined as the sequence of activities take place instantly between an economic event *e* and the event-related investment decision *D*. That is, an information cycle does not exist in a periodical information request environment. Information cycles are found in a timely or "near timely" information request environment. The complete cycle time Δt of the sequence, by our definition, must be shorter than a very small tolerable interval to capture the essence of "timeliness"[6]. Within each cycle, it is reasonable to assume *q* always follows *e* since the event is changing the current economic state. Likewise, "*Rau-AO*" and "*t-Cau*" are assumed to have the similar onto, instant and sequential relations as "*e-q*". However, as well as *D* is assumed to always follow *q*, the information disclosure or auditing activities after *q* are not promised. Two types of information cycle are illustrated as follows.

Perfect information cycle is the information cycle containing timely web-release *dr* and the necessary audit technology between *q* and *D*. For example, information cycle $C_i = (e_i - q_i - dr_i - RAu_i - AO_i - D_i)$ is the perfect information cycle adopting the *real-time auditing* technology, and $C_j = (e_j - q_j - dr_j - AO_j - D_j)$[7] is the perfect information cycle adopting the *continuous auditing* technology. *Imperfect information cycle* is the information cycle containing no timely web-release *dr* or the necessary audit technology between *q* and *D*. In the case of "no-disclosure" type of imperfect cycle, we use $(e_i - q_i - x - x - x - D_i)$[8] to demonstrate the *disclosure inefficiency* caused by the missing subsequence "$dr_i - RAu_i - AO_i$". And $(e_j - q_j - x. - x - D_j)$ to demonstrate disclosure inefficiency caused by the missing subsequence "$dr_j - AO_j$" and the missing continuous auditing work "$t - CAu$". In the case of "no-audit" type of imperfect cycle, we use ($e_i - q_i - dr_i \ -x - x- \ D_i$) to demonstrate the audit inefficiency caused by the absence of "$RAu_i - AO_i$", and ($e_j - q_j - dr_j \ -x- \ D_j$) to demonstrate the audit inefficiency caused by the missing opinion "AO_j" and the "$t -x$".

A company can select its web-release method by two presumed disclosure approaches: the *Pushed-by-Company* (Push) approach and the *Pulled-by-Market* (Pull) approach. The push method means the company itself determines the timing, content and format of disclosure regardless of the market's information demand. Conversely, the pull method provides the "information-on-demand" disclosure mechanism by which the company can generate the newest information with respect to the market's "browsing" requests. Although the pull approach is essentially capable of satisfying all users' information needs, these two approaches are assumed to be indifferent, while the perceptions for the materiality of occurring events are agreed between the host-company and the market. However, if the information provider and the receiver do not have the same perception over the materiality issue, then the pull approach is expected to achieve higher disclosure efficiency than the push method.

Disclosure efficiency measures the capability of the *disclosure approach* adopted by one company for meeting the market's information needs completely and timely. By notation, a disclosure approach is efficient if for each information cycle C_i, it contains a "pull" dr_i right after "$e - q$" and "D_i", or it contains a "push" dr_i right after "$e - q$" and "D_i" on the premise that the company and the market have the same perception over the materiality of events.

Audit approaches are identified by their tests-performing and opinion-issuing frequency and timing. *PAu* denotes the traditional periodical auditing. Real-time Auditing (*RAu*) ensures that each C_i containing dr_i will be a perfect cycle by performing auditing technology $RAu_i - AO_i$ right after dr_i to provide reasonable assurance before any investors use the web-based financial information to make their decisions. To be more specific, in *RAu* approach, the audit work and the audit report are triggered by and performed right after dr_i , and will be completed by an instant batch processing "$RAu_i - AO_i$". Continuous auditing (*CAu*) is the audit approach which ensures that each C_i with disclosure efficiency dr_i will be a perfect cycle by performing continuous auditing technology *CAu* on a transaction basis and providing reasonable assurance AO_j right after dr_j before any investors use the web-based financial information to make their decisions. To be more specific, in *CAu* approach, the audit work CAu_j is triggered by the recognition of every book transaction t_j, and the audit report will quickly summarize the results of CAu_j once the dr_j "pulled" by the market. Therefore, *CAu* will create two required sequences of different frequency. One is the audit work sequence "$t_j - CAu_j$", which has the same frequency as transactions' occurrence. The other is the audit report sequence "$dr_j - AO_j$", which has the same frequency as the web-release.

Audit efficiency measures the capability of the audit approach adopted by one company for completely and timely assurance of the information released. By notation, an audit approach is efficient if for each dr_i , there follows "$RAu_i - AO_i$" (in the case of the real-time auditing) or "AO_i" (in the case of the continuous auditing which audit work is done on a transaction basis) right after "dr_i" and before "D_i". The real-time auditing and the continuous auditing approaches are efficient auditing while the traditional auditing approach is not since it cannot meet the "timeliness" criteria.

An information cycle is *informational efficient* only if it adopts both the efficient disclosure and an efficient audit approach. By definition, only the pull-method perfect information cycle and the push-method premise disclosure efficiency. Table I summarizes the instances by concepts used in the conceptual framework.

2.4 Scenario analysis and hypothesis

Based on prior research, several audit efficiency indicators are considered other than disclosure efficiency and audit efficiency in comparing various audit approaches: audit quality (e.g. Chi *et al.*, 2009; Francis and Yu, 2009; Hay and Davis, 2004), audit cost (Krishnan *et al.*, 2008; Palmrose, 1989), expected audit failure loss (Dye, 1991; Palmrose, 1988), internal control improvement and continuous monitoring (e.g. Chou *et al.*, 2007; Murthy, 2004; Vasarhelyi and Halper, 1991). All efficiency indicators are labeled from I1 to I8 to measure the performance of various auditing approaches (see Table II).

Based on the proposed conceptual framework, we develop a $3 \times 2 \times 2 \times 2$ matrix to compare the economic efficiency for three auditing approaches (periodical auditing, real-time auditing and continuous auditing) under the joint-combination of various disclosure types (push vs pull), materiality perceptions (with common knowledge vs. without common knowledge) and information environments (timely vs periodical). After a reasonable reduction, 12 scenarios (labeled as M1 to M12) are given respectively as presented in Table III. Note that in Table III, it is assumed, in a periodical information environment, the material level of a single event will not influence the capital market's belief and therefore only periodical disclosures exist.

The disclosure efficiency, audit timeliness and audit completeness for each scenario are further analyzed and illustrated in Figure 1. In the preliminary analysis, the scenarios adopted continuous auditing with the pull disclosure (M9 and M11) are most efficient for any given real-time information environments.

As Figure 1 shows, continuous auditing is far better than other audit approaches as long as its expected cost can be controlled. This conclusion motivates us to develop a not costly real-time or continuous auditing technology to test the following hypothesis. Ceteris paribus, given the appropriate technology, the total economic welfare under continuous auditing will never be less than the real-time auditing, and the real-time auditing will never be less than the traditional periodical auditing, regardless of the information environment type.

3. Technical framework

The goal of our technical framework is to develop a set of event-driven programmed components, called the *external continuous audit machine* (ECAM), deployed on the auditor's secured server to provide the following basic *automated* functions:

- to detect the system control configurations through an "*OLCT*" methodology;
- to use the control testing result to determine the nature, timing and extent of *continuous substantive tests*;

Concepts	Instances
Information request environment	TIRE/PIRE
Information cycle (IC)	Perfect IC/Imperfect IC
Disclosure approach	Push/Pull
Disclosure efficiency	Disclosure efficiency/disclosure inefficiency
Audit approach	Periodical auditing/real-time auditing/continuous auditing
Audit efficiency	Audit efficiency/audit inefficiency
Information efficiency	Information efficiency/information inefficiency

Table I.
Instances by concepts in the conceptual framework

Table II. Summary of efficiency indicators

Indicator		Ref. No.	Definition	Related research
Disclosure efficiency		I1	The capability of the disclosure approach adopted by the company to completely and timely meet the market's information needs	This paper
Audit quality		I2	The capability of discovering and reporting material misstatement of accounting information	Chi *et al.* (2009); Francis and Yu (2009); Hussainey (2009); Sundgren (2009); Hay and Davis (2004); Dye (1991); Palmrose (1988); DeAngelo (1981)
Audit cost		I3	The auditor's time and resource in conducting an audit work	Krishnan *et al.* (2008); Palmrose (1989)
Expected audit failure loss		I4	The expected value of loss due to the audit failure	Palmrose (1988); Dye (1991)
Audit efficiency (timeliness)		I5	Audit efficiency is promised for any given information cycle	This paper
Audit efficiency (completeness)		I6	Audit efficiency is promised for each information cycle during a given period	This paper
Other complementary characteristics	Internal control improvement	I7	The internal control improvement function because of auditing	Ettredge *et al.* (2006); Simunic and Stein (1987)
	Continuous monitoring	I8	The real-time and ongoing monitor of unusual anomalies in the transaction processing	El-Masry and Reck (2008); Chou *et al.* (2007); Murthy (2004); Vasarhelyi and Halper (1991)

Table III.
Scenarios design

		Periodical auditing Push disclosure	Periodical auditing Pull disclosure	Real-time auditing Push disclosure	Real-time auditing Pull disclosure	Continuous auditing Push disclosure	Continuous auditing Pull disclosure
Timely information environment	With common knowledge	M1		M5		M9	
	Without common knowledge	M2	M3	M6	M7	M10	M11
Periodical information environment	With common knowledge	M4		M8		M12	
	Without common knowledge	Not applicable (It is assumed, in a periodical information environment, the material level is always a common knowledge between market and company)					

- to perform the *online substantive testing* on a transaction basis; and
- to summarize the substantive testing results and issue an adjusted report simultaneously accompanied by the company's web-releases.

Instead of using an embedded audit module, which can execute the substantive testing periodically only, ECAM emphasizes three automated distinguishing characteristics: Internet connected, online control testing and continuous substantive testing. As Figure 2 illustrates, the "waterfall" system development approach was used to separate the whole technical framework into system analysis and system design/implementation phases.

In this section, the three generic analysis processes required to accomplish the ECAM's framework are described in detail. The first two processes will enhance the planning of ECAM and the OLCT methodology. Using the revised Internal Control Description Language (ICDL) proposed by Bailey *et al.* (1985), an OLCT model is developed to continuously detect the system application control configurations. Afterwards, the third process will construct the technical framework of CSTM which contains 13 transaction-triggered auditing procedures.

3.1 Planning of ECAM

Compare the two different situations appearing in Figure 3, we can see that situation 2 (i.e. the OLCT case) can obtain the updated control configuration data timely upon any changes of system controls. In contrast, situation 1 may lose the latest control data so the substantive testing work may be polluted by the wrong knowledge about system controls.

Furthermore, OLCT's testing results are treated as the blueprint for CSTM. Therefore, when OLCT discovers any control weakness from the control configuration data, it would trigger the later substantive testing procedures for this weakness. In other words, the testing procedures must be preset to relate to every testing objective in OLCT including its necessary tests to be performed, data items to be selected and performing time. In the entity-relationship schema, the control objective and its related audit procedures would provide a one-to-many relationship by cardinality constraints.

Three problems need to be clarified in the planning stage of ECAM. First, the testing objectives of both OLCT and CSTM must be identified to highlight the functional directions of ECAM. Second, the audit risks implied by ECAM objectives need to be examined carefully so that the risk exposure of applying ECAM can be reduced. Finally, the specific system purposes for both OLCT and CSTM are

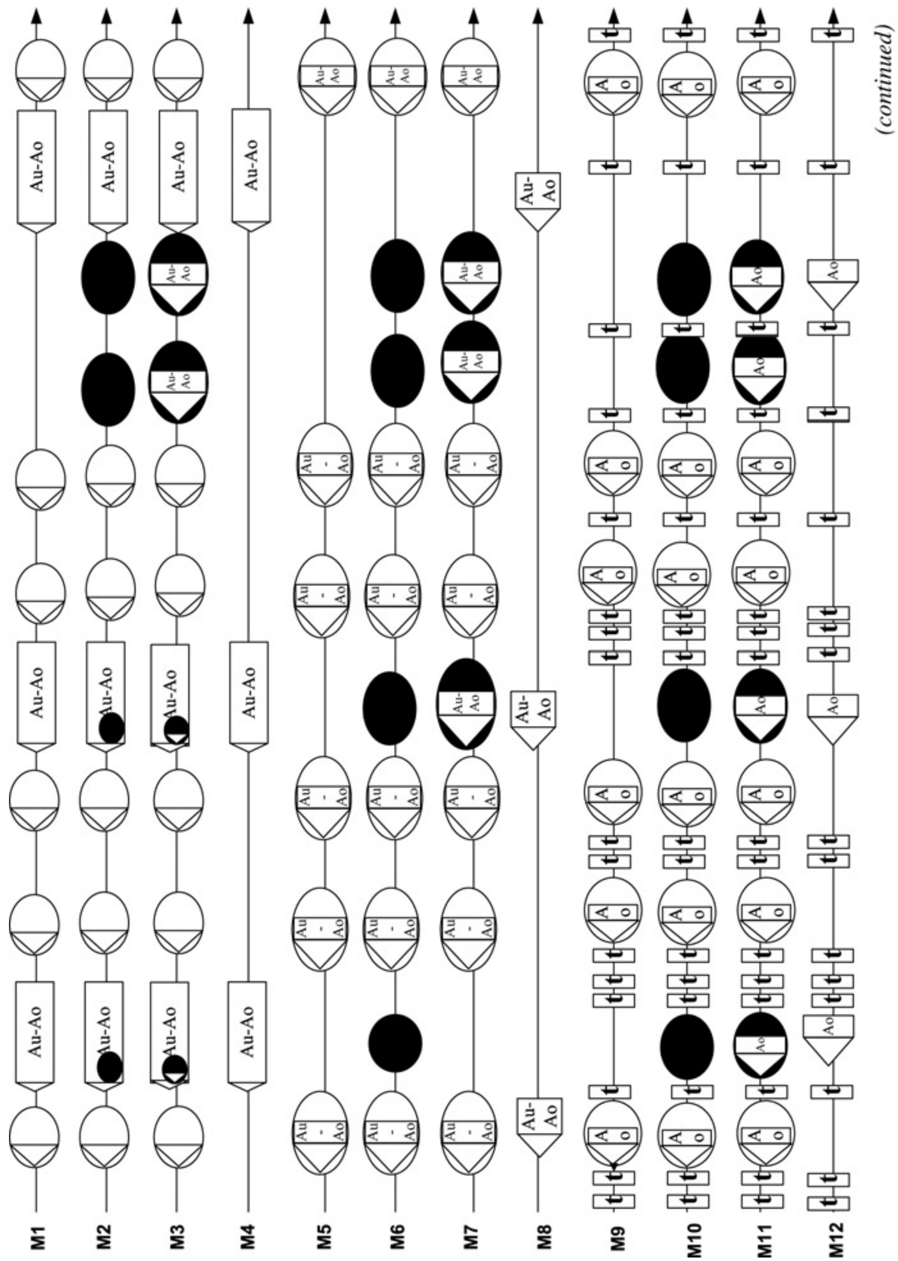

Figure 1.
Illustrations of audit
timeliness and
completeness

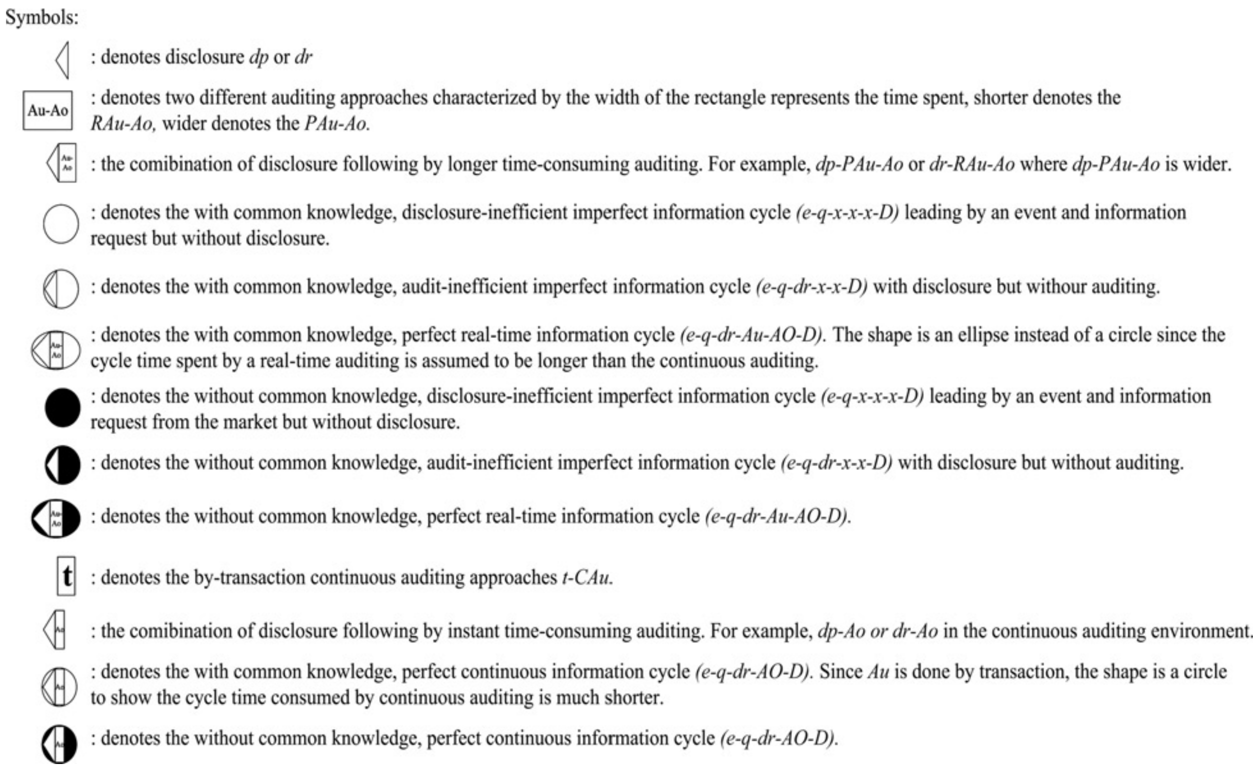
Symbols:
: denotes disclosure *dp* or *dr*
Au-Ao
: denotes two different auditing approaches characterized by the width of the rectangle represents the time spent, shorter denotes the *RAu-Ao,* wider denotes the *PAu-Ao.*
: the comibination of disclosure following by longer time-consuming auditing. For example, *dp-PAu-Ao* or *dr-RAu-Ao* where *dp-PAu-Ao* is wider.
: denotes the with common knowledge, disclosure-inefficient imperfect information cycle *(e-q-x-x-x-D)* leading by an event and information request but without disclosure.
: denotes the with common knowledge, audit-inefficient imperfect information cycle *(e-q-dr-x-x-D)* with disclosure but withour auditing.
: denotes the with common knowledge, perfect real-time information cycle *(e-q-dr-Au-AO-D).* The shape is an ellipse instead of a circle since the cycle time spent by a real-time auditing is assumed to be longer than the continuous auditing.
: denotes the without common knowledge, disclosure-inefficient imperfect information cycle *(e-q-x-x-x-D)* leading by an event and information request from the market but without disclosure.
: denotes the without common knowledge, audit-inefficient imperfect information cycle *(e-q-dr-x-x-D)* with disclosure but without auditing.
: denotes the without common knowledge, perfect real-time information cycle *(e-q-dr-Au-AO-D).*
t
: denotes the by-transaction continuous auditing approaches *t-CAu.*
: the comibination of disclosure following by instant time-consuming auditing. For example, *dp-Ao or dr-Ao* in the continuous auditing environment.
: denotes the with common knowledge, perfect continuous information cycle *(e-q-dr-AO-D).* Since *Au* is done by transaction, the shape is a circle to show the cycle time consumed by continuous auditing is much shorter.
: denotes the without common knowledge, perfect continuous information cycle *(e-q-dr-AO-D).*

Figure 1.

determined from the previous identified testing objectives and audit risk model. The planning activities for OLCT and CSTM are described in the following subsections.

3.1.1 Planning activities for OLCT. There are two planning activities for OLCT: identifying the control testing objectives and assessing control risks. The major mission of OLCT is to obtain the control testing results for determining the substantive testing objectives in CSTM. In our framework, the focus of OLCT is on application controls[9] which protect specific transaction processing functions. The configuration data of these application controls are usually computerized in application programs. For example, when the sales invoice amount fails to be set as a programmed formula by summarizing the extensions of quantity and price for each product item, the OLCT can discover this poor prerecorded control from the control configuration file and create the related balance-testing of invoice amounts. The configuration data of other system application controls, such as the input validity control, input accuracy control and input completeness control, can be assessed automatically through the same way described above. Therefore, OLCT's mission is to determine the strength of those configured application controls adopted by a client's system.

The second planning activity is risk assessment. Formula 3-1 is OLCT's risk assessment model derived from the standard audit risk model.

Formula 3-1. Develop the optimal OLCT technology x, so that $x \ni \hat{CR} \cong CR$ and $P(\hat{CR} < CR) \rightarrow 0$.

From Formula 3-1, we know that the advantage of using OLCT is to lower auditor's risk of underestimating control risks, denoted as $P(\hat{CR} < CR)$. Underestimating control risks may cause audit failure. Therefore, the performer of continuous auditing must obtain the best control testing technique, such as OLCT, to reduce $P(\hat{CR} < CR)$. The following five propositions are success factors for OLCT.

P1. When the configuration data of system control is available, theoretically, the auditor may obtain the best estimator $\hat{CR} \cong CR$ by comparing the realized control configuration data to the standard configurations. If the mapping result is perfect, the auditor's conclusion will be $\hat{CR} \cong CR \cong 0$, which means no further substantive tests are needed since the perfect system control can reduce the overall audit risk approximately to zero.

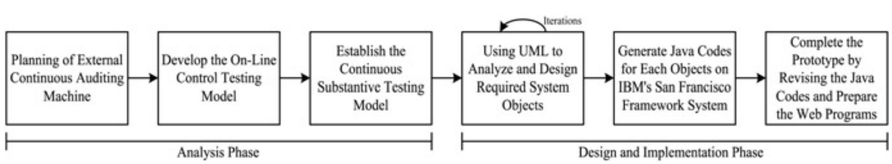

Figure 2. System development process of ECAM

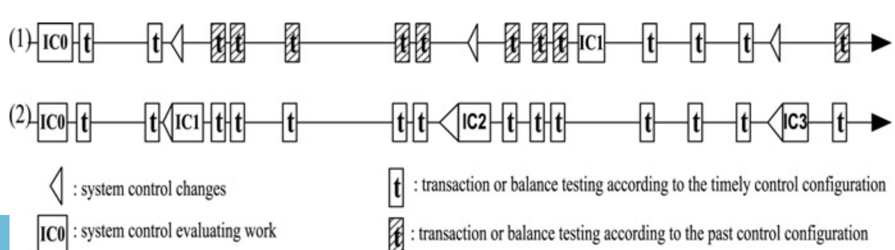

Figure 3. Online system control testing illustration

P2. Another common situation of $\hat{CR} \cong CR$ occurs when the mapping result cannot be perfect due to control weaknesses. In this circumstance, the auditors still can obtain a perfect estimator where $0 < \hat{CR} \cong CR < 1$, but further substantive testing would be required to reduce the detection risk, as well as the overall audit risk.

P3. If the configuration data are not totally available, then the auditor might need to rely on the vendor's certified application components or performing additional black-box control testing methods to reduce $p(\hat{CR} < CR)$. Otherwise, the unpredictable effects caused by $p(\hat{CR} < CR)$ might bring the continuous audit failure.

P4. When the control configuration is unavailable, the auditor must set $\hat{CR} = 1$ and rely on the continuous substantive tests completely to reduce/control audit risk to an acceptable level.

P5. For a specific client's system, the data collection mechanism of control configuration appeared in *P1* and *P2* must be maintained for ongoing configuration updates.

3.1.2 Planning activities for CSTM. There are two planning activities for CSTM: identifying the audit objectives and assessing audit risks. Basically, the assertions conveyed by all the transaction data recorded in a client's information system should be the testing objectives. The main criterion for CSTM objectives is the feasibility of facilitating the substantive testing process through computer programs.

Formula 3-2 indicates that the detection risk can be minimized by well-designed substantive testing technologies such as CSTM, under the assumption of no overestimation of the strength of a client's internal controls. Note that the inherent risk is assumed to be constant

Formula 3-2. Selecting the optimal continuous substantive testing technology y to:

$$\min_{y} \mathrm{Exp}(AR) = \tilde{IR} \times \hat{CR} \times DR, \quad \text{where } DR \text{ is } f(y)$$

where AR is the overall continuous audit risk, $\tilde{IR}$ is the inherent risk which is set to be a constant, CR is the true system control risk, $\hat{CR}$ is the expected system control risk suggested by OLCT, DR is the detection risk of continuous auditing which depending on the CSTM technology.

CSTM obviously needs to be designed as a container of various reusable audit patterns[10] composed of audit rules and audit procedures derived from both GAAP and GAAS. Accordingly, the following two propositions are necessary for the success of implementing CSTM.

P6. For a specific client's system, the CSTM must develop event-triggered audit patterns, including all audit rules, audit procedures and transaction data retrieval mechanism to ensure the continuum of transaction-based substantive testing. The audit pattern will be tightly related to the outcomes of the control configuration to reduce the detection risk DR.

P7. For a specific client's system, the transaction data collection mechanism in *P6* must be maintained for the on-going data schema updates. For the auditor's overall client base, this maintenance facility also needs to ensure its reusability from one client to another.

Summarizing the results of planning activities, we may conclude there are several factors that auditors need to consider while designing an effective OLCT and CSTM. For OLCT, the successful factors are:

- the complete understanding of "standard" control configuration;
- the continuing availability of control configuration data;
- the maintenance and reusability of OLCT mechanism; and
- the reliability of a client's application components.

The first factor concerns the auditor's ability in developing the "best practice" of system controls. In spite of its importance, the success of this factor depends on the auditor's professional knowledge and general experiences which are beyond the scope of this study. Therefore, we only discuss the other three technical factors.

For CSTM, the successful factors are:

(1) the complete setting of automated audit components;

(2) the continuing availability of transaction data; and

(3) the maintenance and reusability of CSTM mechanism.

The strengthening of all three factors is expected to lower the auditor detection risk.

3.2 Online control testing model

In this section, we first adopt ICDL (Bailey *et al.*, 1985) to establish the methodology for identifying the best practice of control configuration and providing an overall system control evaluation model. This evaluating technique will also be used to link each control configuration to its substantive testing procedures. In other words, it will create a dynamic model for OLCT's control activities. Furthermore, the static conceptual model of OLCT will be constructed on the object-oriented technical foundation to address the data schema problem.

3.2.1 System control evaluating model. The processing of one transaction is always triggered by a set of events. These triggering events are usually the post-conditions of precedent transactions before a specific principal node. All of the preconditions and post-conditions[11] of each transaction will form a transaction network. Over the network, ICDL defines the precedent constraints (PCs)[12] to identify the preconditions for each principal transaction node[13]. On the left top of Figure 4, we illustrate a fragment of the network constituted by transaction nodes and their PC sets.

By a further thinking of PC, it is possible to create a network map of standard PCs to evaluate a client's present control configurations. The standard PC sets, denoted as {PC*}, are the "best practice" of system controls. Any deficiency from the standard map can be considered a weakness of application control. Therefore, we may simply introduce the following linear evaluation model to address the idea.

$$\text{DIST\_TOTAL} = w1 * (\text{DIST1}) + w2 * (\text{DIST2}) + w3 * (\text{DIST3})$$

In this model, DIST_TOTAL measures the total difference between the expected and the realized control configurations[14]. In Table IV, we describe the three main terms, DIST1, DIST2 and DIST3, representing the collections of different levels of

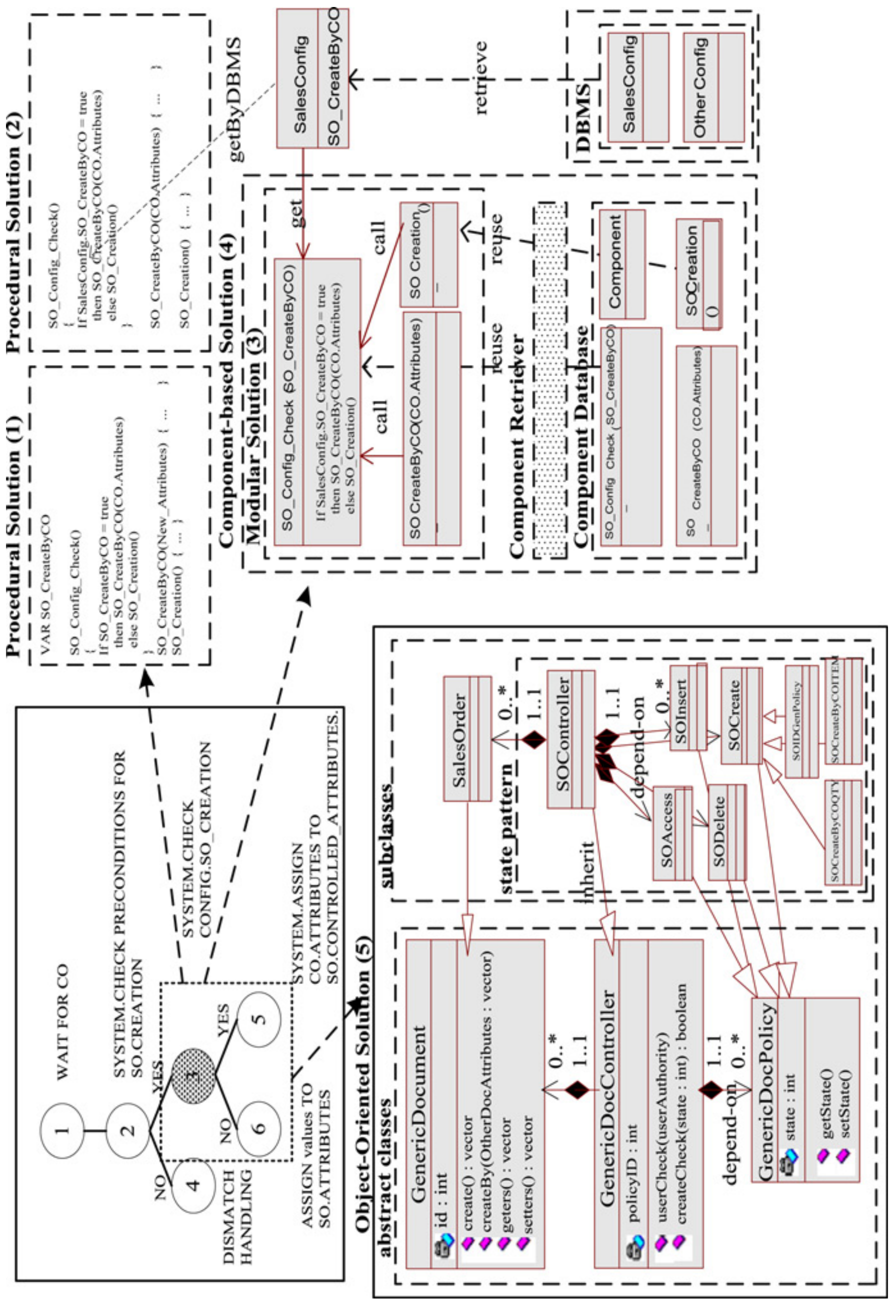

Figure 4.
Illustration of five types of OLCT designs

Risk measures	Description
DIST1	DIST1 stands for the least deficient situation that we call "inconsistency". Using ICDL words, DIST1 collects the inconsistent deficiencies described as follows: "For each (n_l, r_k) in {PC} under auditing, it is found a corresponding pair $(n_l, r_k)^*$ in {PC*} and each n_l in (n_l, r_k) will be identical to n_l^* in $(n_l, r_k)^*$. However, there exists some r_k is not equal to r_k^*." For example, the (M-2) application control plan appeared in Table IV requires any new SO should be tightly related to customer order (CO) and the product data entity. Therefore, one standard control should assign SOPrice = ProductStandardPrice ±5%. If the test result of SOPrice is the assigning of SOPrice = ProductStandardPrice ±25%, it will constitute a relatively slight accuracy problem which should be included in DIST1
DIST2	DIST2 is the moderate case of deficiency that we call "incomprehensive" deficiency. Using ICDL terms, DIST2 is the case when each n_i in {PC} has an identical node ni* in {PC*}, there exists some r_k^* in {PC*} but $r_k^* \notin$ {PC}. One example of DIST2 is that (M-2) does not implement any presetting formula for SOPrice. If it is the case, the auditor might find out larger variance on sales prices to decrease the accuracy of transaction data than DIST1
DIST3	The worst situation is the "incompleteness", represented by DIST3, which means there exists some n_l^* {PC*} but $n_l^* \notin$ {PC}, as well as its related preconditions r_k^*. We may use the SO creating node in Table IV as an illustration of DIST3. Restricted by (M-1) and (M-2), any new SO must be inherited from an unrecorded CO. If (M-1) and (M-2) were not implemented, which means SO creating node is not related to any post-conditions of CO, DIST3 exists in the SO creating node. This deficiency might increase the possibility of fictitious transactions so that a serious further investigation on the existence assertion might be necessary

Table IV.
Descriptions of OLCT risk measures

weaknesses. Different weights on materiality ($w1$, $w2$ and $w3$) are assigned to the three levels of deficiency. We may expect $w1 < w2 < w3$. Those deficiency collectors are assumed to inclusively represent all types of control configurations[15]. After screening the three deficiency indicators, auditors may obtain the overall control configuration performance DIST_TOTAL.

The main purpose for OCLT to evaluate system application controls is to induce the prerequisites for the following substantive testing. Therefore, in addition to summarizing the overall control risk value, a more crucial goal is to specify the required testing procedures for auditing those deficiencies. Through careful examination of the control matrix highlighted by Gelinas and Dull (2008), Table V, with a slight extension, demonstrates a chronological form of auditor's {PC*}. These control matrix tables are used to break through the {PC*} map for identifying detail substantive testing. As Table V illustrates, the last column lists the necessary substantive testing rules or procedures relating to the control weakness (the "M" label) in the order entry processing.

3.2.2 Requirements for OLCT Methodology. Generally, the data of control configurations addressed by ICDL or control matrix could be collected through widely accepted system control testing methods if there is no time constraint. However, when the transaction processing systems become more complex and more reliable (Elliott, 1998, 1995), these methods are sometimes too time-consuming and unnecessary. Since time factor is crucial for the usefulness of OLCT, new technology for obtaining complex control configuration data needs to be developed. Figure 4 illustrates five general types of design approaches and data models:

Table V.
Control matrix of sales order initiating, permitting and maintenance

Present and missing control plans[a] (triggering preconditions)	Operational control goals: Ensure operational effectiveness	Ensure operational efficiency	Ensure the security of resources	Control goals of information processing: For SO inputs: IV	IC	IA	For SO updates: UV	UC	UA	Auditing	Impacts on substantive test if audit plan is not implemented[b]
P-1: Logs of 4 Ws when any user logs in to input/update any records in SO										P	Need not to do further substantive tests (Mandatory)
M-1: SO creation condition Checking (1. user's authorization; 2. one-to-one related to UnRecorded Customer Order;)	M		M	M	M						(1) check if UserID is legal; if not, start CORRECTIVE process; (2) check if SO is one-to-one related to CO; if not, start CORRECTIVE process;
M-2: SO tightly inherited from CO and Product: (1. SOLineItemQuantity = COLineItemQuantity; 2. SOPrice = StandardPrice ± 5%; etc.)	M			M	M	M	M	M	M	M	(3) check if SOLineItems = COLineItems; if not, start CORRECTIVE process; (4) check if SOLineItemQuantity = COLineItemQuantity; if not, start CORRECTIVE process; (5) check if SOPrice between StandardPrice ±5%; if not, start CORRECTIVE process;
M-4: SO Permission Condition Checking (1. user's authorization; 2. ToPermit SO; 3. querying on SO permission policies)	M		M	M							(6) check if there is permission process; if not, start CORRECTIVE process; (7) check if UserID is legal; if not, start CORRECTIVE process;
M-5: SO Permission Policy Setting (1. CreditLimitPolicy isNotNull; 2. CreditLimitPolicy. CreditLimit > SOAmount; 3. SOLineItemQuantity = COLineItemQuantity; 4. SOPrice between StandardPrice ± 5%; etc.)	M									M	(8) check if CreditLimitPolicy Null; if Null, auditor creates the standard CreditLimitPolicy; (9) check if CreditLimitPolicy.CreditLimit > SOAmount; if not, start CORRECTIVE process;

(*continued*)

Present and missing control plans[a] (triggering preconditions)	Operational control goals: Ensure operational effectiveness	Operational control goals: Ensure operational efficiency	Operational control goals: Ensure the security of resources	Control goals of information processing: For SO inputs IV	IC	IA	For SO updates UV	UC	UA	Auditing	Impacts on substantive test if audit plan is not implemented[b]
M-6: Quick response between: 1. CO creation and SO creation; 2. SO creation and permission; 3. SO update notice and SO update		R									(10) need not to do further substantive tests; (11) check if there is update process; if not, start CORRECTIVE process;
M-7: SO update condition checking (1. user's authorization; 2. ToUpdate SO.CorrectionNotices exists; 3. querying on SO policies)	M		M				M	M	M		(12) check if UserID is legal; if not, start CORRECTIVE process

Notes: [a]The control plans are labeled by P (present) or M (missing), representing the testing results for each (n_1, r_k*) pair. Their achieving control goals are given in the next seven columns, also labeled as P, M or R (recommended); [b]Those testing procedures simply include the "detecting" and "correcting" procedures for the possible misstatement caused by "non-preventive" control weaknesses. For example, if (M-2) is found not implemented by {PC}, which means the price and quantity might be inconsistent with the data records listed on CO and Product. This may cause an inaccurate record on sales revenue. Therefore, the testing procedure for (M-2) weakness will trace each unpaid sales record back to CO and Product. If the sales amount is, found by the detecting procedure, not equal to the extension of COQuantity and ProducetStandardPrice, the correcting procedure will adjust back to the accurate amount

Table V.

(1) the procedural system;
(2) the procedural system with control data managed by RDBMS;
(3) the modular system with RDBMS;
(4) the component-based system with RDBMS; and
(5) the object-oriented system consists of various state patterns with RDBMS.

In Table VI, we compare the comparative advantages on the three criteria for the five design approaches that the client system might take.

3.2.3 Influence of Client System on OLCT. The first criterion requests OLCT to ensure the availability of three types of configuration data for any given transaction in the client system:

(1) the precedent nodes of the principal transaction;
(2) the expression of each constraint; and
(3) the current state of each constraint.

However, these data are usually mined in the system logic, program variables or transaction databases. Thereby, it is necessary to examine if the client system could guarantee the timely retrieval and transmission of those data. We introduce the "workflow control"[16] method for auditors to evaluate the availability of client system's control data. This idea is similar to the separation of the PC from the principal transaction node in ICDL. For example, on the left top of Figure 4, we present a fragment of sales order processing using a revised ICDL. In this example, node 3 (*n*3) checks the creating constraints of sales order (SO) before any SO creation. If the post-condition of *n*3 shows that SO_CreateByCO is true, then *n*5 will be triggered, else *n*6 will be triggered. In this case, the control flag SO_CreateByCO is obviously the key configuration data for SO creation. Depending on SO_CreateByCO, auditors will know exactly the control strength for SO creation by identifying the outcome of the control flag instead of analyzing the program logic. Table VI concludes that while OO method is taken by the client, as illustrated in Figure 4, auditors may obtain the best estimate of control risk through the separate control configuration class.

The responsibility for continuously monitoring of control changes must be reinforced in OLCT because the "continuity" concept raised by continuous auditing should not only emphasize on the assurance of historical events, but also on the present and future events. Three types of changes regarding the control of any given transaction node are analyzed. The first is the state change of constraints[17]. The second is the add, delete and update (*ADU*) processing of the related constraints of precedent nodes[18]. The third is the add and delete (*AD*)[19] processing of the principal transaction's precedent nodes[20]. A traditional procedural system will have to update, recompile the whole program and it is difficult for auditors to find out where and what the change has been made. In contrast, it is easier for auditors to specify the updated or replaced flow-control in a modular system by monitoring the latest update time in an "indexed" program library and check the updates in a relatively smaller piece of program. In the component-based system, auditors can rely on the component management system to identify whether the replacement of flow control makes the system control better or worse.

Whereas the control configuration data availability concerns the degree of "coupling" among transaction nodes, the reliability of application components concerns

Table VI. Possible control and detection risk induced by various system design approaches

	Client system			ECAM			
System design approach	Availability of control configuration data[a]	Maintenance of control data retrieval	Transaction data accessibility	Reliability of system components	Maintenance of OLCT and CSTM mechanism	Reusability of OLCT and CSTM mechanism	Continuing retrieval of transaction data and control data
Pure procedural	Low	Low	Low	Low	Low	Low	High
Procedural system with DBMS	High	Low	High	Low	Low	Low	High
Modular system	High	Medium	High	Low	Medium	Low	High
Component-based system	High	High	High	Medium	High	Medium	High
Object-oriented system	High	High	High	High	High	High-embedded inheritance mechanism	High

Notes: [a]Among the five design approaches appeared in Figure 4, only the pure procedural system cannot provide the SO_CreateByCO since the system logic and control data cannot be appropriately separated. As illustrated on the fragment of pseudo program in Figure 4, the pure procedural system cannot separate r_3 from n_3. Therefore, both the control configuration expression and current value of r_3 cannot be obtained. The procedural system using database still "implode" n_3, n_5 and n_6 in the same procedure. But it can obtain the current value r_3 from the control configuration data table. The modular and component-based systems can further "explode" n_3, n_5 and n_6 into different modules or components. However, the expression of r_3 is still "mined" in the program. Finally, the OO method can retain the current value of r_3 in the attributes (e.g. SOCreateByCOQTY, SOCreateByCOITEM, etc.) of "policy" objects (e.g. SOCreateByCOQTY instantiated from SOCreate class which inherited from GenericDocPolicy superclass). The manipulating expression of r_3 are coded into specific methods (e.g. getSO_CreateByCO(), updateSO_CreateByCO(), etc.) and be encapsulated in the same object, waiting for the "controller" object SOController to check the current states. Furthermore, different creation methods (e.g. SOCreate(), SOCreateByCO(), etc.) are restored in the SalesOrder object and labeled by the object identities (OIDs) to be exactly retrieved by SOController. Apparently, the OO method can achieve the perfect "explosion" of transaction nodes, expressions of control configuration and the current states of control configuration. Therefore, the OO method can promise the most desirable control information and reduce the risk of using OLCT the degree of "cohesive" control for each transaction process (Parnas, 1972; Stevens *et al.*, 1974). Since auditors might wonder that even SO_CreateByCO is collectible, how can auditors be sure that the SO creating procedures are reliable? Although *P1* and *P2* are the ideal situations for OLCT, usually the complete knowledge about "cohesive" controls is impossible. Therefore, the information of "cohesive" controls usually comes from a reliable software vendor's evaluation or the observed disciplined system development process. Figure 5 suggests an independent information provider of system reliability. The "certification authority" (CA) is assumed to be generally acknowledged by software vendors and auditors for evaluating system reliability. CA is responsible for the authentication of software vendor identities, the approval and issuance of certificates for their software components, and the maintenance of each vendor's certificate information in the public database. When a software vendor submits the applications for a certification, a CA agent should review the vendor's quality controls over software development and decide whether to approve the applications. After careful examination, the CA can rank the applicant's system to the appropriate class according to the reliability of system application functions. Note that the application control reliability is quite different from the system reliability generally defined in software engineering. Application reliability will pay more attentions to the availability, security, integrity and maintainability (referred to AICPA's SysTrust^{TM}) of application functions rather than general system functions. Any control changes are required to reregister with the CA, and the CA also has the responsibility to review the reliability information periodically. Auditors or other interested users are allowed to access those data through Internet as Figure 5 illustrates.

The existence of CA, appeared in Figure 5, for providing system reliability information can substantially reduce auditor's risk of using OLCT especially when the client system is continuously ranked as the "AAA" (highest reliability) class. However, before the practice of CA, auditors need to acquire a general knowledge about client system reliability through a theoretical way. In general, ceteris paribus, the well-

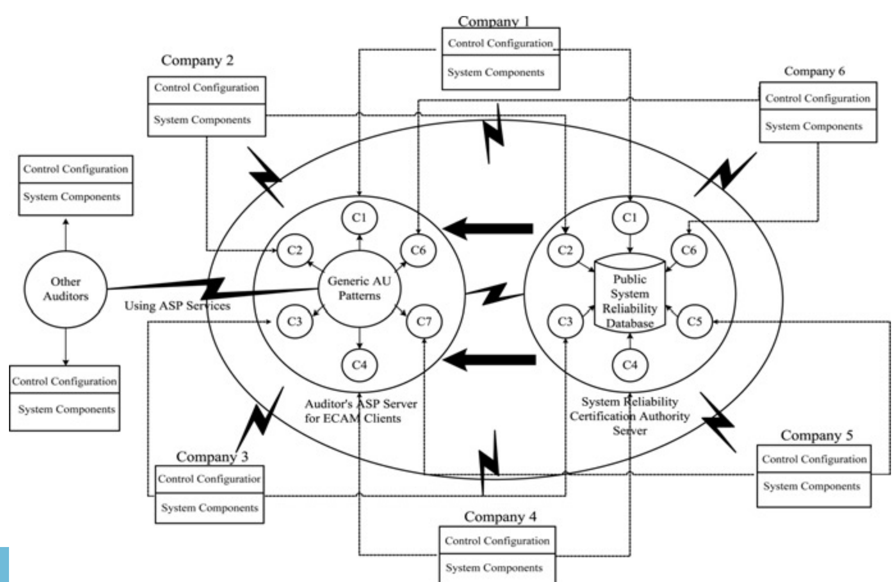

Figure 5.
Suggested ASSP framework of ECAM

designed component-based approach and OO approach will create the most reliable application components since the robustness of these two system approaches have been proved by many software engineering theoretical literatures.

3.2.4 Data model requirements for OLCT. Some of the successful factors for OLCT, such as the availability of control configuration, also depend on OLCT's "event-sensor"[21] mechanism. Therefore, OLCT must maintain a timely online data retrieval mechanism to capture the ongoing control changes and maximize reusability of the mechanism. We specify several necessary conditions for a well-designed OLCT in Table VI.

Recall that OLCT is required to collect three types of configuration data for any given transaction:

(1) the precedent nodes of the principal transaction;

(2) the expression of each constraint; and

(3) the current state of each constraint.

In addition to identifying the successful factors endowed in the client system, it is also necessary to develop an efficient OLCT method to ensure the timely retrieval and transmission of those data. Apparently, program tracing is the most direct method to obtain those data, but no applicable technology can perform the "timely" control logic tracing. Another choice is the "black-box" method that has been broadly used in traditional testing procedures. Nonetheless, it is also unacceptable to implement a human-intensive online sensor of system control.

What we need is an automated data retrieval gateway along with an "event-sensor" in the front end to detect and capture each new event continuously on the given resource locations. On the server side, control configuration database and control mapping mechanism are required to store the data of {PC}, {PC*}, and the mapping of {PC} and {PC*}. The effectiveness of core facilities for the control data retrieval, storage and mapping solely depends on the audit knowledge, database tools and Internet technology, not on the system development approaches. So the solution of mechanism will be left to the design and implementation phase.

OLCT mechanism needs to continuously provide the control data retrieval, storage and security. As many literatures indicate (Bohem, 1981; Pressman, 1999; Booch *et al.*, 1999a, b), OO method is preferred because of its superior adherence in developing and managing software components. Another concern is when the auditor's business expands, whether the prior developed OLCT technology can be reused in new contracts? It is undoubted that software reuse is always the most important feature of OO method (Booch *et al.*, 1999a, b). Reusability can be applied not only in document control, but also in process management and in audit knowledge. Considering that the high reusability of audit components can increase the auditor's competitive advantage, an assurance service provider model of ECAM is suggested in Figure 5. In ASSP model, an auditor firm with highly developed ECAM can "rent" its ECAM or other assurance service components to other auditors to gain the advantage of reusable components.

3.3 Continuous substantive testing model

Recall that the CSTM is required to create the best continuous substantive testing components for reducing the detection risk (*P6*) and to ensure the ongoing effectiveness of those components (*P7*). Basically, the influence of client system design approach to CSTM is relatively smaller than it is in OLCT since transaction database usually are deployed in a separate tier in a multi-tier client-server architecture. Therefore, as long as the application tier can feed in the complete transaction data to the database tier or the application interface tier, the accessibility of transaction data would be guaranteed. But if auditors adopted OLCT as a front tier prior to CSTM, the design of client system still will influence the accuracy of CSTM indirectly, as well as the case of bad control risk estimate inducing higher detection risk.

Similar to OLCT, CSTM requires an automated transaction data retrieval gateway along with an "event-sensor" to continuously detect and capture each "posted" but "not tested" transaction from given resource locations. A macro container of the records of retrieved raw transaction data, performed testing procedures and tick marked testing results is needed for collecting, analyzing and restoring the information of follow-up substantive tests. Basically, just like the prior arguments for OLCT components, the effectiveness of the transaction data retrieval and testing mechanism also depends on the audit knowledge, database tools and Internet technology. However, the maintaining of the effectiveness will be differed under different design approaches. Again, OO method is recommended for CSTM as shown in Table VI due to its better endowment in software robustness, flexibility and reusability.

3.4 Conceptual framework of ECAM

After considering all of the requirements, we propose a conceptual framework of ECAM consisting of both OLCT and CSTM. Thirteen general processes interactively flowing among various ECAM components and client system modules are defined in Figure 6. Therefore, any design approach capable to exercise the generic concepts addressed in the framework will be a feasible technical solution to ECAM.

Step (1) appeared in Figure 6 indicates the process OLCT retrieving control configuration data from a client's system. Steps (2) and (3) show how OLCT compares the client's control outcome to the standard control template. With the data retrieval gateway, step (4) captures transaction data on an instant basis to trigger the necessary audit procedures identified in step (5), based on the setting of the control configuration. The testing and correcting results of CSTM's audit procedures are accumulated in the auditing test database as step (6) illustrates. CSTM also presets the segmental materiality for presenting the important transaction anomaly to the auditor in step (7). The detection of timely web-releasing report by step (8) immediately triggers step (9) to summarize the accumulated testing and correcting results and produce an adjusted report. According to the preset overall materiality, in steps (10) to (13), CSTM issues a suggested continuous audit report near simultaneously attaching on the same page of the client's release.

4. Conclusions

This paper contributes to the solution of auditing web-released financial information by presenting a conceptual framework of continuous auditing and developing conceptual framework of an ECAM. The conceptual framework simulates various information disclosing and auditing environment and argues that the continuous auditing would be the most appropriate approach for web-releasing assurance. Although the hypothesis derived from the framework still needs further empirical supports, the anticipated sustaining is reasonable under the emergent web-release practice.

In considering the continuous auditing technology, the concept of OLCT is strongly promoted. In the OLCT model, we claim that well-controlled workflows can pass the subsequent transaction testing procedures, which significantly raises the "synergy" between internal control assessment and substantive testing. Of course, this benefit is

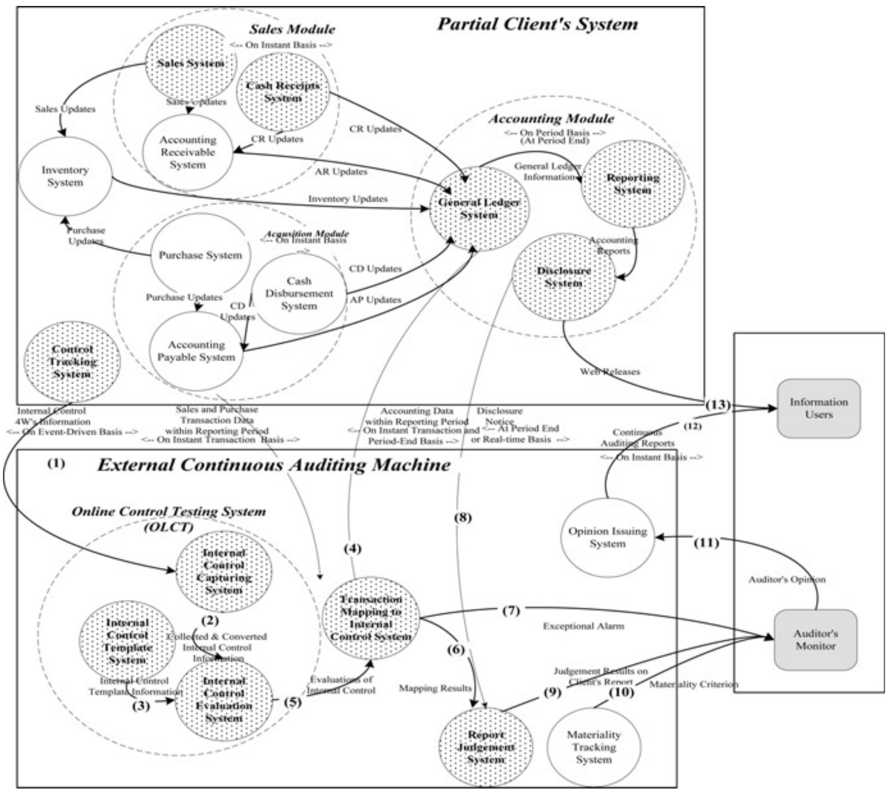

Figure 6.
Conceptual framework of ECAM

ensured through OO approach by separating the control objects (controllers), the control configuration data (policies) and the general operation objects (documents). According to the moving trend of software engineering (Brereton *et al.*, 1999), more robust business operational components are anticipated. Therefore, we suggest a CA responsible for continuously issuing and retaining various types of certificates to software vendors for their general business components. Further expanding the model of collaborating with the CA, the continuous auditors can joint venture with other parties or assurance service providers (e.g. WebTrust, SysTrust and other certification authorities, etc.) to gain high-quality outsourcing services. In the near future, when more companies adopting the emerging XBRL technology in transforming transaction data, it will be possible to provide the most reliable data format for continuous auditors to perform remote and automated continuous auditing.

Notes

1. eXtensible Business Reporting Language (XBRL) could be the most appropriate for the preparation and exchange of global business reports and data (www.xbrl.org/).
2. By the word "generic", we simply mean no software component design or specific application details are involved, only conceptual functions or components will be captured in the generic analysis.

3. Economic events are independent to each other in the real world. However, the total information requests for serially related events will not be greater than the same number of the independent events. Therefore, for simplicity, the information requests for occurring events are assumed to be mutually independent, so are the following information requests and information disclosures.
4. "Timely auditing" means auditors can finish audit tasks and issue the audit report right after company's financial information disclosure. See CICA and AICPA, 1999.
5. It should be noticed that, in the real world, no such an environment would exist.
6. The meaning of tolerable level of "timeliness" can be referred to the Figure 2 in the Continuous Audit Report (AICPA and CICA, 1999).
7. There is no audit work contained in the cycle, since the audit work is done on a transaction basis. Which means the audit work "$t - CAu$" is done after each t, not in an event-driven information cycle.
8. "x" stands for the absence of activities in the event-driven sequence expected to adopt continuous auditing.
9. ISACA (2000) defines "application controls" as those relate to the transactions and standing data appertaining to each computer-based application system and are therefore specific to each such application.
10. Gamma *et al.* (1995) defines the *design pattern* as a set of reusable artifacts providing a specific function.
11. The set of preconditions and post-conditions for a transaction can be considered as the "contract" of transaction processing. In the later discussion, we will show that UML also uses "contract" to define system behavior.
12. The PC of one principal transaction node i are formally defined as a set of precedent transaction states to trigger the principal, which could be restated as $\mathrm{PC}_i = \{(n_1, r_k)$ | if and only if node i follows n_1 under the preconditions r_k, where (n_1, r_k) follows a one-to-many relationship$\}$. By definition, each transaction involved in the processing system will be assigned its PC set and becomes one member node on the whole network {PC}.
13. This paper adopts the object technology to develop a different data schema, and embeds methods into objects. Therefore, the E-R model and the command set are not applied in our analysis.
14. The control risk is assumed to be the increasing function of DIST_TOTAL.
15. This statement could be simply proved by a mathematical combination under the assumption of one-to-many relationship between the precedent node and its preconditions for triggering the principal node.
16. The earliest work flow methodology was developed by IDEF Users Group and National Institutes for Standards and Technology (NIST), known as the IDEF3 product derived from the Integrated-Computer-Aided-Manufacturing DEF project (see www.idef.com/). IDEF3 creates a work flow model which attempts to isolate the control logic from the work processes.
17. State change is the most common case for control change. For example, assume the acceptable price limit for new sales orders is preset to be within ProductStandardPrice ± SOPriceQuota, and the value of SOPriceQuota attribute is reassigned from 5 percent to a looser range, say 15 percent. State change can be auto-detected since the criteria for collecting instant value is similar to the availability criterion. Therefore, OO method is preferred than others as concluded in the prior section.
18. This scenario indicates the situation when the expressions of certain constraints change but the basic structure of {PC} network still remains unchanged. For example, assume the original price policy is "if SalesPrice is not within ProductStandardPrice ± SOPriceQuota,

then not accept", but now a new policy is added, e.g. "if SalesPrice is less than ProductStandardCost, then not accept".

19. The updates of precedent nodes are not included because they are identical to the constraint updates.
20. Sometimes the PC_i nodes and their constraints will be entirely removed or appended. For example, the client might cancel the entire n_3 and reassign the SO creation process to n_6. This is a more complex scenario since the AD processing will not only involve with the constraints but also the nodes for operation. Simply indexing on programs cannot guarantee the simultaneous processing of operating functions and controlling functions, so the procedural and modular system will both be inappropriate for auditors to collect the update information. Similarly, the high-cohesion and low-coupling component-based system is expected to be the moderate approach for such updates because of its component management system.
21. Generally, two kinds of Internet "event-sensor" approaches can be found in the search engine technology. One is called "pull" method, and the other is "push" method. "Pull" method is auditor's server will be facilitated as the web crawlers which use some "daemon" or "intelligent agent" programs established by, e.g. the "setTimeout" mechanism in Java, to frequently check if there are new updates in some given URLs. "Push" method will rely on the event-triggering mechanism in client's system to self-initiate the update data delivery to auditor's server over the Internet.

References

AICPA and CICA (1999), *Continuous Auditing*.

APB (1970), "Basic concepts and accounting principles underlying financial statement of business enterprises", APB Statement No. 4, par. 40, AICPA, New York, NY.

Bailey, A.D., Duke, G.L., Gerlach, J., Ko, C., Meservy, R.D. and Whinston, A.B. (1985), "TICOM and the analysis of internal controls", *Accounting Review*, Vol. 60 No. 2, pp. 186-200.

Bohem, B.W. (1981), *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ.

Bohrer, K.A. (1998), "Architecture of the San Francisco frameworks", *IBM Systems Journal*, Vol. 37, pp. 156-62.

Booch, G.J. and Rumgaugh, I.J. (1999a), *The Unified Modeling Language User Guide*, Addison-Wesley, Reading, MA.

Booch, G.J. and Rumgaugh, I.J. (1999b), *The Unified Software Development Process*, Addison-Wesley, Reading, MA.

Brereton, P., Budgen, D., Bennett, K. and Munro, M. (1999), "The future of software", *Communications of the ACM*, Vol. 42 No. 12, pp. 78-84.

Chambers, A.E. and Penman, S.H. (1984), "Timeliness of reporting and the stock price reaction to earnings announcements", *Journal of Accounting Research*, Vol. 22, pp. 21-47.

Chi, W., Huang, H., Liao, Y. and Xie, H. (2009), "Mandatory audit partner rotation, audit quality, and market perception: evidence from Taiwan", *Contemporary Accounting Research*, Vol. 26 No. 2, pp. 359-91.

Chou, C.L., Du, T. and Lai, V.S. (2007), "Continuous auditing with a multi-agent system", *Decision Support Systems*, Vol. 42, pp. 2274-92.

CICA (1972), *CICA Handbook*.

Collins, D.W. (1994), "Lack of timeliness and noise as explanations for the low contemporaneous return-earnings association", *Journal of Accounting & Economics*, Vol. 18, pp. 289-324.

DeAngelo, L. (1981), "Auditor independence, 'Low Balling', and disclosure regulation", *Journal of Accounting and Economics*, Vol. 4, pp. 113-27.

Dye, R.A. (1991), "Informationally motivated auditor replacement", *Journal of Accounting and Economics*, Vol. 14, pp. 347-74.

Elliott, R.K. (1995), "The future of assurance services: implications for academia", *Accounting Horizon*, Vol. 9 No. 4, pp. 118-27.

Elliott, R.K. (1998), "Assurance services and the audit heritage", *Auditing: Journal of Practice and Theory*, Vol. 17, pp. 1-7.

El-Masry, E.E. and Reck, J.L. (2008), "Continuous online auditing as a response to the Sarbanes-Oxley Act", *Managerial Auditing Journal*, Vol. 23 No. 8, pp. 779-802.

Ettredge, M.L., Li, C. and Sun, L. (2006), "The impact of SOX Section 404 internal control quality assessment on audit delay in the SOX era", *Auditing: A Journal of Practice and Theory*, Vol. 25 No. 2, pp. 1-23.

Ettredge, M.L., Simon, D., Smith, D. and Stone, M. (1994), "Why do companies purchase timely quarterly reviews?", *Journal of Accounting & Economics*, Vol. 18, pp. 131-55.

FASB (1979), "Financial accounting and changing price", Statement of Financial Accounting Standards No. 33, FASB, Stamford, CT.

Francis, J.R. and Yu, M.D. (2009), "Big 4 office size and audit quality", *The Accounting Review*, Vol. 84 No. 5, pp. 1521-52.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, MA.

Gelinas, U.J. and Dull, R.B. (2008), *Accounting Information Systems*, 7th ed., South-Western College Publishing, Boston, MA.

Hay, D. and Davis, D. (2004), "The voluntary choice of an auditor of any level of quality", *Auditing: A Journal of Practice and Theory*, Vol. 23 No. 2, pp. 37-53.

Hussainey, K. (2009), "The impact of audit quality on earnings predictability", *Managerial Auditing Journal*, Vol. 24 No. 4, pp. 340-51.

ICAEW (1975), *The Institute of Chartered Accountants in England and Wales Members, Handbook*.

ISACA (2009), *COBIT and Application Controls: A Management Guide*, available at: www.isaca.org/Template.cfm?Section=browse_by_category&CONTENTID=52294&TEMPLATE=/ContentManagement/ContentDisplay.cfm

Krishnan, J., Rama, D. and Zhang, Y. (2008), "Costs to comply with SOX Section 404 auditing", *Auditing: A Journal of Practice and Theory*, Vol. 27 No. 1, pp. 169-86.

Murthy, U.S. (2004), "An analysis of the effects of continuous monitoring controls on e-commerce system performance", *Journal of Information Systems*, Vol. 18 No. 2, pp. 29-47.

Palmrose, Z. (1988), "An analysis of auditor litigation and audit service quality", *Accounting Review*, Vol. 53, pp. 55-73.

Palmrose, Z. (1989), "The relation of audit contract type to audit fees and hours", *Accounting Review*, Vol. 64, pp. 488-99.

Parnas, D.L. (1972), "On the criteria to be used in decomposing systems into modules," *Communications of the ACM*, Vol. 15 No. 12, pp. 1053-58.

Pressman, R.S. (1999), *Software Engineering*, 4th ed., McGraw-Hill, New York, NY.

Searcy, D.L. and Woodroof, J.B. (2003), "Continuous auditing: Leveraging technology", *The CPA Journal*, Vol. 73 No. 5, pp. 46-8.

Simunic, D.A. and Stein, M.T. (1987), "Product differentiation in auditor choice in the market for unseasoned new issues", The Canadian Certified General Accountants' Research Foundation, Burnaby.

Sinclair, N.A. and Young, C.Y. (1991), "The timeliness of half earnings announcement and stock returns", *Journal of Accounting and Finance*, pp. 31-52.

Stevens, W.P., Myers, G.J. and Constantine, L.L. (1974), "Structured design", *IBM Systems Journal*, Vol. 13 No. 2, pp. 115-39.

Sundgren, S. (2009), "Perceived audit quality, modified audit opinions and the likelihood of liquidating bankruptcy among financially weak firms", *International Journal of Auditing*, Vol. 13 No. 3, pp. 203-21.

Vasarhelyi, M.A. and Halper, F.B. (1991), "The continuous audit of online systems", *Auditing: A Journal of Practice and Theory*, Vol. 10 No. 1, pp. 110-25.

Verrecchia, R. (1983), "Discretionary disclosure", *Journal of Accounting and Economics*, Vol. 5, pp. 179-94.

Verrecchia, R. (1990), "Information quality and discretionary disclosure", *Journal of Accounting & Economics*, Vol. 12, pp. 365-80.

Zeghal, D. (1984), "Timeliness of accounting reports and their informational content on the capital market", *Journal of Business Finance & Accounting*, Vol. 11, pp. 367-80.

Further reading

Columbus, L. (2000), *Realizing e-Business with Application Service Provision*, Sams Technical Publishing, Indianopolis, IN.

Datar, S.M., Feltham, G.A. and Hughes, J.S. (1991), "The role of audits and audit quality in valuing new issues", *Journal of Accounting and Economics*, Vol. 14, pp. 30-49.

Fayad, M.E. and Schmid, D.C. (1997), "Object-oriented application frameworks", *Communications of the ACM*, Vol. 40 No. 10, pp. 32-8.

Givoly, D. and Palmon, D. (1982), "Timeliness of annual earnings announcements: some empirical evidence", *Accounting Review*, Vol. 57, pp. 486-508.

Groomer, S.M. and Murthy, U.S. (1989), "Continuous auditing of database applications: an embedded audit module approach", *Journal of Information Systems*, Vol. 3 No. 2, pp. 53-69.

Hoitash, R., Hoitash, U. and Bedard, J.C. (2008), "Internal control quality and audit pricing under the Sarbanes-Oxley Act", *Auditing: A Journal of Practice and Theory*, Vol. 27 No. 1, pp. 105-26.

Kogan, A., Sudit, F. and Vasarhelyi, M.A. (1996), "Implications of Internet technology: on-line auditing and cryptography", *IS Audit & Control Journal*, Vol. 3, pp. 42-7.

Larman, C. (1998), *Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design*, Prentice-Hall, Englewood Cliffs, NJ.

Pastena, V. (1979), "Some evidence on the SEC's system of continuous disclosure", *Accounting Review*, Vol. 54, pp. 776-83.

About the authors

Chi-Chun Chou is Assistant Professor of Accounting at National Taipei College of Business. He earned his doctorate from The National Cheng-Chi University in Taiwan. Chi-Chun Chou has studied issues in accounting information systems with the emphasis on XBRL technology, ontological engineering and electronic auditing. His papers have appeared in *Expert System with Applications*, *International Journal of Intelligent Systems in Accounting, Finance and Management*, *Sun Yat-Sen Management Review*, *Soo-Chow Journal of Accounting*, *Securities and Futures Monthly*, among others. Chi-Chun Chou is also the leading academic counselor of Financial Supervisory Commission for XBRL adoption in Taiwan. Chi-Chun Chou is the corresponding author and can be contacted at: ccchou412@gmail.com

C. Janie Chang is the Vern Odmark Professor of Accountancy at San Diego State University. She earned her doctorate from The University of California at Irvine. C. Janie Chang has studied issues in auditing, accounting and information systems with the emphasis on information processing of experts and cross-cultural issues related to professional judgments and decisions in auditing and managerial accounting. Her papers have appeared in *Abacus*, *Auditing: A Journal of Practice and Theory*, *Behavioral Research in Accounting*, *Data Base*, the *International Journal of Accounting*, *Journal of Accounting Literature*, *Journal of Business Ethics*, *Journal of Multinational Financial Management*, *Journal of Public Budgeting*, *Accounting and Financial Management*, *Review of Business Information Systems*, among others.