CONTINUOUS AUDITING TECHNOLOGIES AND MODELS

by

Adrian W. Blundell

CONTINUOUS AUDITING TECHNOLOGIES AND MODELS

by

ADRIAN BLUNDELL

DISSERTATION

Submitted in the fulfilment of the requirements for the degree

MAGISTER TECHNOLOGIAE

in

INFORMATION TECHNOLOGY

in the

FACULTY OF ENGINEERING, THE BUILT ENVIRONMENT AND INFORMATION TECHNOLOGY

of the

NELSON MANDELA METROPOLITAN UNIVERSITY

Supervisor: PROFESSOR R. VON SOLMS

Co-supervisor: S. FLOWERDAY

January 2007

Declaration

I, Adrian Wesley Blundell, hereby declare that:

- The work in this dissertation is my own work.
- All sources used or referred to have been documented and recognized.
- This dissertation has not previously been submitted in full or partial fulfilment of the requirements for an equivalent or higher qualification at any other recognized educational institution.

_____

Adrian Wesley Blundell
10 January 2007

Acknowledgements

I would like to thank the following:

- My supervisors, Mr Stephen Flowerday for your guidance, support and patience, and special thanks to Prof R von Solms.

- Tracey, my love, you encourage and inspire me.

- My family, I appreciate the encouragement and support (notably financial). Thanks also to the Du Plessis family, who also showed me great encouragement.

- The financial assistance of the National Research Foundation (NRF) is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the author and are not necessarily attributed to the NRF.

- Most of all, God, through Whom all things are possible.

Abstract

Continuous auditing is not a totally new concept, but it has not been widely implemented, and has existed mostly as a point of debate amongst the auditing fraternity. This may soon change, as continuous auditing has become a topic of great interest, especially in the last decade. This may be due to a combination of reasons. In the last decade, much of the confidence in auditors' reports was lost due to corporate governance scandals. This also brought about a greater desire for faster, more reliable reporting on which to base decisions. This desire has been transposed into regulations such as the Sarbanes-Oxley act in the United States, which encourages real-time auditing activities, which would benefit from continuous auditing. A second, possible contributing factor to the heightened interest in continuous auditing is that much of the requisite technology has matured to a point where it can be successfully used to implement continuous auditing. It is the technologies which form the focus of this research. It is therefore, the primary objective of this research to investigate and identify the essential technologies, and identify and define their roles within a continuous auditing solution. To explore this area, three models of continuous auditing are compared according to the roles of the technologies within them. The roots of some auditing technologies which can be adapted to the paradigm of continuous auditing are explored, as well as new technologies, such as XML-based reporting languages. In order to fully explore these technologies, the concepts of data integrity and data quality are first defined and discussed, and some security measures which contribute to integrity are identified.

An obstacle to implementing a continuous model is that even with the newly available technologies, the multitudes of systems which are used in organisations, produce data in a plethora of data formats. In performing an audit the continuous auditing system needs to first gather this data and then needs to be able to compare "apples with apples". Therefore, the technologies which can be used to acquire and *standardise* the data are identified.

Table of Contents:

List of Figures and Tables:

PART I – Background Chapters

Chapter 1

Introduction

1.1 Prologue

An article on continuous auditing of database applications was published as early as 1989 (Groomer & Murthy, 1989). Despite this there still exists a wide variety of opinions as to what a continuous audit actually is; and there are very few practical examples of implemented systems (Vasarhelyi, 2002).

A typical definition of continuous auditing is as follows:
"a comprehensive electronic audit process that enables auditors to provide some degree of assurance on continuous information simultaneously with, or shortly after, the disclosure of the information" (Rezaee, Sharbatoghlie & McMickle, 2002).

Some definitions suggest continuous auditing is used by internal auditors only, and others refer to the external audit process (Alles, Kogan & Varsarhelyi, 2004). It may also be viewed to encompass both internal and external auditing (ISACA Standards Board, 2002).

Although there are many definitions of continuous auditing, they all agree in two ways:

1) Firstly, the aim of the audit is to provide assurances of one sort or another.

2) Secondly, the nature of the reports is that they are produced as soon as possible after the events on which they are based.

According to literature, it can also be established that in order to rapidly produce results, as is desirable in continuous auditing systems, highly automated electronic systems need to exist (Shields, 1998). A continuous auditing system relies on a flow of information in a fully automated process. Thus, technology plays a pivotal role in continuous auditing. However, many of the technologies required to perform

continuous monitoring of real-time accounting systems has only recently become sufficiently available and cost efficient (Alles, Kogan & Varsarhelyi, 2005).

1.2 Description of Problem Area

In the modern era of real-time accounting systems, real-time reporting has become more desirable, in order to provide decision makers with timely information. Also fuelling this desire is the need to comply with legislation, such as Section 409 of the Sarbanes-Oxley Act, which has come about after corporate governance scandals such as Enron (Alles et al., 2004). However, the intended research will not emphasize these legal and governance issues. This research will rather focus on the technological environment and specifically technologies which enable continuous auditing, and to this end, three of the published theoretical models of continuous auditing will be examined.

An area of particular interest is that of information/data quality and integrity. One of the main purposes of continuous auditing is to assist auditors in providing assurances on financial reports by verifying information integrity (Flowerday and von Solms, 2005). The conclusions in auditors' reports must be based on accurate and reliable data in order to be trustworthy (Wessmiller, 2002). Therefore, technology must be implemented with an understanding that the integrity of data within the system is also of the utmost importance, as the integrity of underlying data may later affect the integrity of information produced by the system. For this reason, the concepts of information/data quality and integrity must first be introduced, and contextualised according to this research, before the main area of focus can be explored.

The main area of focus concentrates on the technical aspects of continuous auditing. This area has not received as much attention as other aspects of continuous auditing, such as legal/regulatory aspects, in published literature. This focus area will examine some technologies which may be regarded as essential to the creation and functioning of a continuous auditing system. These include:

- Technologies such as Expert Systems and embedded audit modules.
- XML-based technologies, such as XBRL (eXtensible Business Reporting Language), XARL (eXtensible Assurance Reporting Language) and XCAL (eXtensible Continuous Auditing Language).

Some tools which can assist auditors, have existed in one or another form almost since the advent of computers, however as the auditing paradigm changes towards continuous auditing, use of these tools needs to adapt to new auditing requirements. Bearing this in mind, the software and methodologies known to auditors and accountants as Computer Aided Tools and Techniques (CAATTs) will need to be examined, as well as the history of Electronic Data Processing (EDP) and Auditing. How all of the aforementioned tools fit together to achieve the aims of continuous auditing in their newly expanded roles will be explored, as well as where newer technologies, such as XML-based technologies, are required.

1.3 Problem Statement

Many technologies exist to aid the auditor in his/her duties; these have to be adapted to assist in meeting the aims of continuous auditing. Furthermore, new technologies may also be required, and these may be relatively unknown.

1.4 Research Objectives

The purpose of this research project is to identify technologies which are essential to continuous auditing and establish how these are used, together, within continuous auditing models to provide continuous assurances.

1.4.1 Primary Objective:

The primary objective of this dissertation is to investigate and identify technologies which assist in supporting continuous auditing and thereby provide continuous assurances. These technologies will then be placed in context, by examining continuous auditing models, and tabulating the roles of technologies within these models.

1.4.2 Secondary Objectives:

In order to accomplish the aims of the primary research objective, it is necessary to first accomplish three secondary objectives. These objectives are:

1) to explore problems surrounding the multitude of different data formats which exist. These formats hinder the flow of data and information, such as transactions, between systems.

2) to clarify the concepts of data integrity and quality. An understanding of these aspects is essential to understanding the roles of certain technologies within continuous auditing models.

3) to introduce and discuss the concepts of accuracy and reliability of data, information and the system, and to show how essential these are to continuous auditing.

1.5 Research Philosophy

The research paradigm leans towards the phenomenological, but has elements of positivistic research (Collis & Hussy, 2003).

1.6 Research Methodology

An extensive literature study forms the basis of this research. Comparisons between models found in literature are tabulated and discussed. The research findings have been documented and disseminated by preparing a paper which has been published in issue 25 of the Journal 'Computers and Security'. This dissertation presents the final collated results of the study (Olivier, 1997).

1.7 Layout of Dissertation

The layout of chapters can be grouped into two main parts. The first part comprises the background chapters, which aim to introduce the reader to some important concepts. An understanding of these concepts is required before the main problem area can be addressed. This part introduces the problem area and the environment in which a solution is required. The second is the solution part, which proposes a solution to the main problem. A third part concluded the paper, and comprises the appendix.

1.7.1 Background Chapters

The first five chapters will discuss the environment in which continuous auditing exists. This includes the technological environment and elements of information security, specifically, integrity and data quality.

The relationships between the auditing and information technology environments will be discussed, as the history of how these two areas developed and intertwined is explored in Chapter Two.

Chapter Three continues to discuss the technological advancements which lead to continuous auditing, and focuses on Computer Aided Tools and Technologies.
Chapter Four looks at a technology which facilitates the adoption of real-time reporting, eXtensible Business Reporting Language (XBRL). The role of XBRL in ensuring data reliability will be examined.

Chapter Five steps away from the technological environment, and looks at data/information quality and integrity, which are important aspects in continuous auditing. These concepts will then be linked to some of the information security aspects, which are crucial to the reliable functioning of a continuous auditing system.

1.7.2 Solution Chapters

Chapters Six and Seven will concentrate on continuous auditing models and show how various technologies come together. Chapter Six introduces the three selected models of continuous auditing, and details how they function. Chapter Seven evaluates and compares the models. Aspects related to data acquisition and the *data format problem* are also discussed. The dissertation is then concluded and summarized in Chapter Eight.

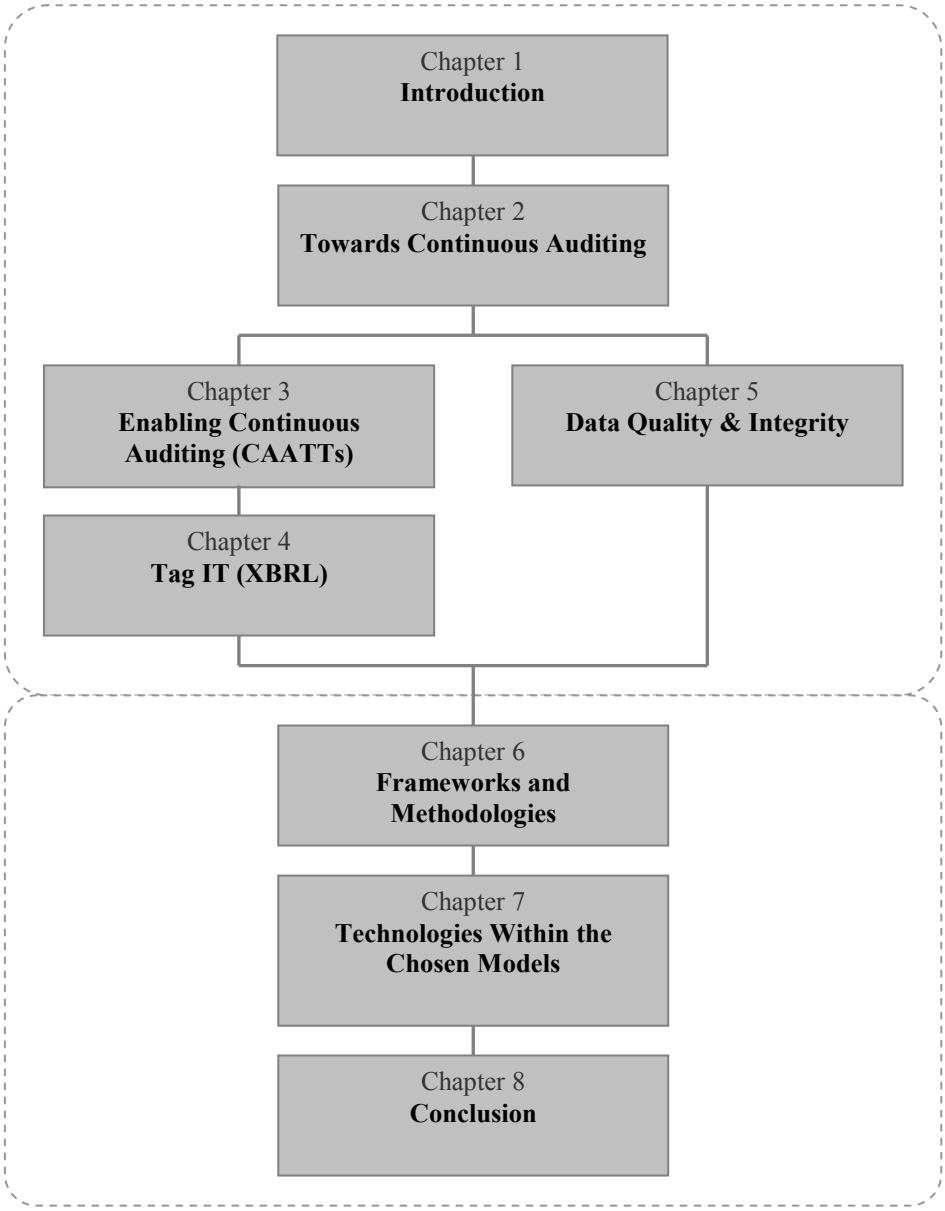
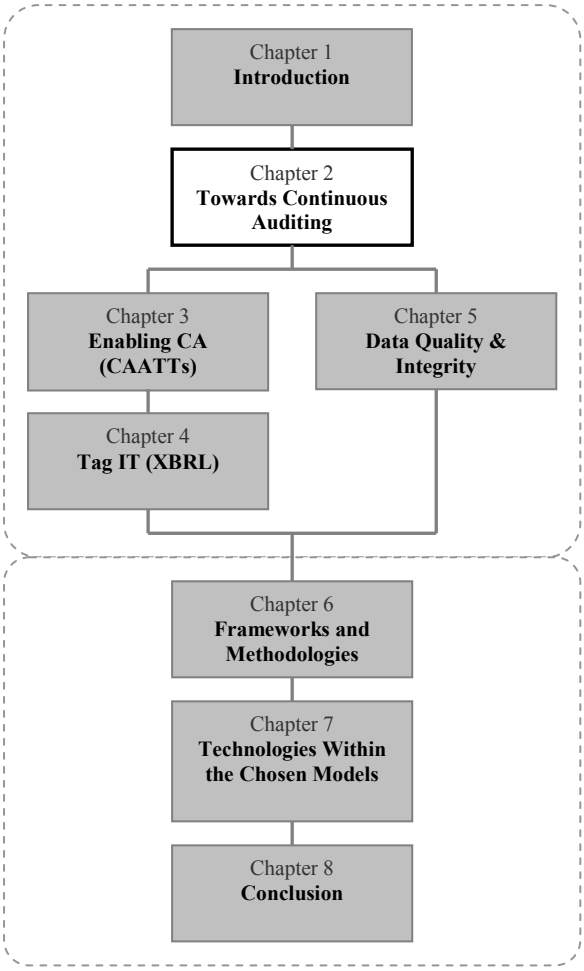


Figure 1.1: Layout of Chapters in Dissertation

Chapter 2

Towards Continuous Auditing



Chapter 2:

2.1 Introduction

2.2 Principles of Continuous Auditing

2.3 Driving Force

2.3.1 History of Electronic Data Processing and Auditing

3.3.2 The Movement Towards Continuous Auditing

3.3.3 Benefits of Continuous Auditing

2.4 Conclusion

2.1 Introduction - A Clearer Picture

This chapter will introduce important concepts relevant to understanding the problem area, in order to create a clearer picture of what continuous auditing is. Continuous auditing will be defined, according to the usage of the term in this dissertation. The aims of continuous auditing will also be established. The definitions of terms similar to continuous auditing will also be clarified. Furthermore, it is important to understand the history of information technology and auditing, and how these developed, together, to drive the desire for continuous auditing. Many of the concepts introduced in this chapter will be built on in further chapters.

2.2 Principles of Continuous Auditing

There are a few definitions of continuous auditing; the most widely accepted is the definition stated in a report by the Canadian Institute of Chartered Accountants (CICA). That definition defines continuous auditing as follows: 'a methodology that enables auditors to provide assurance on a subject matter using a series of auditor reports issues simultaneously with, or within a short period of time after, the occurrence of the events underlying the subject matter' (CICA/AICPA, 1999).

It is important to clarify that continuous auditing is not simply performing traditional audits using technology. Continuous auditing alters the audit process, for instance, there is a change in focus of the auditor. There is an increased focus on adequacy and effectiveness of internal control activities and less prominence is placed on substantive tests of documents and transactions. Manual, traditional audits found evidence of financial misstatements long after they occurred and allowed for little corrective action. Continuous auditing methodologies aim to be more preventative and deterrent of misstatements (Bierstaker, Burnaby, & Thibodeau, 2001; Rezaee, Elam, & Sharbatoghlie, 2001). Continuous auditing also differs from traditional audits in two other ways:

1) As is stated in the definition, the reports are issued at shorter intervals than under traditional audits. Reports may be produced daily or weekly as opposed to annually (Shields, 1998). Alternatively, the reports may be made available immediately, or on demand.

2) The audit can focus on any type of information relevant to decision making, not just financial statements. An example of this is the authenticity, integrity and non-repudiation of e-commerce transactions or the effective operation of controls over a publicly assessable database. Continuous auditing could also report on non financial measures of a company's performance (Shields, 1998).

Continuous auditing is often discussed in literature as a tool for either internal audit or external audit engagements. In this dissertation, continuous auditing will be

considered as a tool for both internal and external auditors. This viewpoint is shared by a 2002 definition in a report by the ISACA Standards Board. Continuous auditing is therein defined as a methodology which allows auditors, external and internal, to issue written reports (ISACA Standards Board, 2002).

There are slight differences in the way internal and external auditors may use continuous auditing. In an article, the definition of continuous auditing as perceived by independent (external) auditors is:

> 'a systematic process of gathering electronic audit evidence as a reasonable basis to render an opinion on fair presentation of financial statements prepared under the paperless, real-time accounting system' (Rezaee et al., 2001).

Continuous auditing is therefore described as a process of gathering and evaluating evidence. This process also aims to establish the efficiency and effectiveness of real-time accounting in safeguarding assets, maintaining data integrity and producing reliable financial information (Rezaee et al., 2001). Continuous auditing, as a tool of internal auditors, aims to put procedures in place to test business processes and management's continuous monitoring process of the control and disclosure environment (Krell, 2004). Business processes are tested by examining large numbers of transactions. Thus the internal auditor would have to firstly examine transactions, and secondly, test the internal control structure of the organisation. The auditor could then provide assurances regarding the quality and credibility of information produced in the real-time accounting environment (Rezaee, Sharbatoghlie, Elam, & McMickle, 2002). Srinivas (2006) elaborates that it is the internal auditors who are the most effective at implementing continuous auditing techniques, because of their knowledge and access to systems. External auditors will also benefit from continuous auditing, as they rely on the work of internal auditors, as well as their own knowledge and expertise (Srinivas, 2006a).

A term commonly referred to in literature is *continuous assurance*. Continuous assurance has been defined in very much the same way as continuous auditing is defined by CICA (Alles, Kogan, & Vasarhelyi, 2004). Continuous assurance therefore refers to the same methodology as continuous auditing, but is perhaps, more specifically the desired end-product of the continuous auditing process.

There are two very different suggested approaches to what is commonly referred to as continuous auditing. The most commonly implemented approach involves constantly monitoring the client's actual system, while the second approach is to reprocess the client's data in a simulated or mirror system, which is monitored (Alles, Kogan, & Varsarhelyi, 2002). This dissertation will be limited in focus to the first approach, as it is closer to the desired ideal described in the CICA report.

A term which also requires clarification is *continuous online auditing*. A 1999 CICA report suggested that continuous auditing is only feasible when implemented as a fully automated process, and that instant access is required to data which details relevant events and their outcomes. The most logical way to satisfy these requirements is if continuous auditing is implemented in an online system. In this context, an online system is where there is a constant network connection between the client's system and the auditor's. Thus, the system is both continuous and online, and it may be described as continuous online assurance/auditing (COA). While COA could be used by both internal and external auditors, because COA requires a relatively intimate knowledge of the client's systems, internal auditors may find it easier to implement COA. COA could also be expensive, due to hardware, software and networking requirements, which would also make it less appealing to external auditors (Kogan, Sudit, & Varsarhelyi, 1999).

Another term mentioned in literature is *continuous monitoring*. Continuous monitoring is not the same as continuous auditing. Continuous monitoring aims to obtain information, for management's use, about the performance of a process, system or data, but it does not aim to produce an audit report like continuous auditing. Therefore, the type of evidence gathered by continuous monitoring is different to that required by continuous auditing. Continuous auditing requires a higher level of proof to produce reports, due to attestation standards. Therefore, it provides information from the auditor's direct personal knowledge, whereas, continuous monitoring merely provides indirect information regarding process, system or data performance (ISACA Standards Board, 2002).

2.3 Driving Force

In order to examine the reasons for a desire and need for continuous auditing, it is necessary to examine the history of auditing and IT. Once this topic has been explored, the benefits of continuous auditing will be discussed.

2.3.1 History of Electronic Data Processing and Auditing

Continuous auditing has its roots in EDP. The first EDP started in the 1950s. Processing was carried out in batch, and the auditor was merely required to compare the machines input to its output - simple parallel processing. Punched cards provided a paper-trail which could easily be traced (Rezaee et al., 2001). When the auditor first calculates an expected set of results and then compares these to the actual results from the system it is known as an 'auditing around the computer approach'. This approach is only effective when the system's application being audited is relatively simple and straightforward (Cerullo & Cerullo, 2003).

In the 1960s computer technology advanced and many companies began adopting computer technology. Computers increased in speed to where online, real-time processing became possible. Tape drives replaced punch cards. This transformed the paper-trail into an electronically stored format. Auditors realized that computers could be used as auditing tools and the first sampling applications came about. The most economical audit method was to use a test deck (test data). Some auditors developed computer programs to help with audit tasks, such as testing mathematical accuracy and comparing files. These are called generalized audit software (GAS). The auditors' mind-set changed towards an 'auditing through the computer approach'(Ramamoorti & Weidenmier, 2004). Whereas 'auditing around the computer' viewed the system as a 'black box', the new approach considered the logic of the system which meant that the auditor would need to have IT experience (Cerullo et al., 2003). Code reviews and other approaches to verify controls and transactions were required, thus computer assisted audit tools and techniques (CAATTs), which are detailed in the next chapter, developed in subsequent years.

In the 1970s mainframe computing was in vogue. EDP auditing was not readily performed in the 1970s and early 80s. Internal auditors were reluctant to make use of

the 'auditing through the computer' methods such as: integrated test facility, tagging and tracing, mapping, parallel simulation, concurrent processing, controlled processing or reprocessing, program code checking and flowchart verification. A possible reason is the high degree of technical skills required to use these tools. Fortunately during the 1980s, when personal computers made accessing data easier, these tools became more user-friendly and required less technical skill for the auditor to implement. The proprietary type of GAS tools used in the early 1970s were replaced by commercially available tools such as ACL and IDEA, which worked across multiple platforms and input file formats (Ramamoorti et al., 2004). These were designed to test automated controls (Cerullo et al., 2003).

The development of IT, as well as the IT Audit is summarized in the table below:

Time Frame	**IT Developments**	**Internal Audit Function developments**	**Evolution of IT Audit**
Mid 1950s	Computer begins processing business applications using punched cards.	(Internal) auditors "audit around the computer."	1st generation EDP Audit: Compliance
1960s	Tape drives replace punched cards Generalized Audit Software emerges	Sampling applications explored Primitive "Auditing through the computer" approach emerges Test decks used to test computerized systems. Internal audit functions begin to perform operational audits	
1970s	25 proprietary GAS packages. ACL created. Multitude of tests created to test computerised systems	IIA issues the influential *Systems Auditability and Control* (SAC) reports	
1980s	Personal computer (PC) is born IDEA software created for PC.	(Internal) auditors continue to slowly experiment with IT	2nd generation IS audit: Control frameworks
1990s	Enterprise Resource Planning (ERP) systems proliferate. Internet use soars. Inter-enterprise integration key to success (CRM, SCM). Ethical Hacking commences. Privacy laws enacted: HIPPA, COPPA, GLBA, Identity Theft and Assumption Deterrence Act.	Internal auditors continue to adapt GAS and expand role within organisations. Rate of IT adoption intensifies with the emergence of the internet.	3rd generation IT audit: Risk/Control COBIT Framework released.

2000s	Internet and global communications technology revolutionise business; computer forensics surges ahead.	Internal audit focuses on supporting Sarbanes Oxley Sec. 302 (CEO/CFO certification), Sec 404 (internal controls management assessment/ auditor/ attestation), and Sec 409 (real-time reporting by issuers).	4th generation IT audit: Risk management process. IT Governance Institute guidance. COSO ERM framework

Table 2.1: Evolution of IT and the Internal Audit Function (Ramamoorti & Weidenmier, 2004, p 347)

2.3.2 The Movement Towards Continuous Auditing

In the 1990s there was a change in the role internal auditors played. This was possibly due to increased demand for value-added services. Value-added services include: improving standardised processes, assisting management with control self-assessments, performing financial function reviews and risk assessments, accessing more information with less disruption to users and improvements in the methods of gathering and analysing the data, which was desired to improve the decision-making process (Glover & Romney, 1997). There were also major technological impacts on auditing which developed during the 1990s, which included:

- More frequent use of word processors and spreadsheets
- Streamlining of Human Resource needs
- Increased use of electronic work papers
- Improved sampling procedures due to more powerful EDP techniques
- Increased communication capabilities (Glover et al., 1997)

Communication capabilities were accelerated by the proliferation of corporate-wide networks, which allowed information to flow throughout the organisation. This meant that organisations implemented Enterprise Resource Planning (ERP) systems to aid in the management of internal processes, such as sales, procurement, human resources, finance, accounting, production, distribution and quality control. The simultaneous emergence of intranets allowed for the sharing and analysis of information internally within the organisation. Extranets then became a way of linking trading partners, blurring organisational boundaries, and this lead to the alignment of technology platforms amongst trading partners. This allowed for an *extended enterprise,* which

facilitated Supply Chain Management (SCM) and Customer Relationship Management (CRM) (Ramamoorti et al., 2004). Another trend was towards e-business, and an entirely new form of business evolved; the native Internet business. At the same time some traditional businesses went through a process coined *electronization* (Vasarhelyi, 2002). The implication is that most transactions occur purely as an electronic process.

One of the most influential communication technologies to gain ground during the 1990s was Electronic Data Interchange (EDI). EDI is a form of Electronic Commerce where trading partners use a specific format to exchange business and financial data between their computer systems. This allows them to conduct their business transactions in a purely electronic environment, which meant that the paper-trails, which had traditionally been relied on as audit evidence, had disappeared (Rezaee & Reinstein, 1998). Source documents that would be used as audit evidence, such as purchase orders, invoices and cheques are replaced by electronic files which are transmitted as electronic messages, as well as the data stored in journal, ledger and schedule files of electronic accounting packages (Rezaee et al., 2001). While the primary objective of the audit remained the same, new auditing procedures were required. Internal auditors were the first to adjust to a lack of paper-trails. In the EDI environment, Internal auditors could no longer rely on substantive tests alone, and had to start testing the effectiveness of controls to gather evidence, which necessitated the adoption of proper safeguards advocated by COSO and COBIT (Rezaee et al., 1998). In order to test controls within a complex IT system a new approach had to be adopted. An 'auditing through the computer' approach developed. This approach focuses on testing: controls, automated processing steps, programming logic and edit routines, since if these are in place irregularities are unlikely to be undetected, and the outputs can be presumed reliable (Cerullo et al., 2003).

During the 1990s, auditors realised that timing of evidence extraction had to change, because some evidence only existing for short periods of time. Audit samples could no longer be gathered at year-end; samples had to be collected throughout the audit period. Continuous auditing suited the needs of Internal Auditors (Rezaee et al., 1998). At the same time electronic information had several advantages: it is more flexible, easier to access, easier to transfer and can be stored, summarized and

organized better than paper-based information. These advantages allowed for the development of real-time accounting systems, which then required the independent (external) auditor to adopt continuous auditing techniques (Rezaee et al., 2001).

As the heightened use of technology forced auditors to obtain evidence electronically, the concept of electronic evidence was incorporated into professional standards (Rezaee et al., 1998). Professional standards such as the Statement on Auditing Standards (SAS) No. 80 by the Auditing Standards Board of the American Institute of Certified Public Accountants (AICPA), which was issued in 1996, suggested the use of continuous auditing, when most of the audited information exists in electronic form (Rezaee et al., 1998).

The discussion on the developments in technology during the 1990s, which facilitated continuous auditing, will be continued in Chapter Two. In the next section the benefits which made continuous auditing attractive will be explored.

2.3.3 Benefits of Continuous Auditing

After continuous auditing became increasingly technologically feasible, many benefits were attributed to it. These benefits include:

- the ability to test a larger data sample, possibly all the transactions. The testing is also faster and more efficient than 'auditing around the computer' techniques (Rezaee et al., 2001).

- reductions to cost and time taken, as compared to a manual audit (Rezaee et al., 2001). Continuous auditing allows for a shorter audit cycle (Srinivas, 2006a).

- the quality of audits may increase, because some of the tedious sampling and analysis is automated, the auditor can focus on gaining an understanding of the client's business and internal control structure (Rezaee et al., 2001).

- an increased flexibility in the audit process, and reports for third parties and client's are more customizable (Srinivas, 2006a).

Perhaps one of the main incentives for adopting a continuous auditing methodology is that compared to traditional audits, continuous auditing aims to be preventative and deterrent of misstatements, rather than merely helping to correct misstatements in financial statements long after the underlying events have occurred (Rezaee et al., 2001). As Bierstaker, Burnaby and Thibodeau (2001, p. 163) state, 'the focus of the audit will shift from manual detection to technology-based prevention'. The prevention of financial misstatements, through the verification of information integrity in audit reports, has become increasingly desirable. This may be attributed to scandals which highlighted a lack of corporate governance and accountability, such as Enron, Worldcom, Tyco and Parmalat, which shook the stakeholders' trust in financial reports (Flowerday & R. von Solms, 2005a). Continuous auditing would have been able to identify problems such as those in Enron, by providing assurances on processes which may not be reported on in the published financial report. Problems would be detected much sooner than in a well-performed traditional audit, and in the case of Enron, the unreported related-party partnerships would have easily been detected. A good continuous auditing system should also report the detected anomalies to a supervisory authority. This would become a deterrent to fraud schemes relating to related-party transactions, overlapping management, double dipping, conflicts of interest and insider trading (Varsarhelyi, Kogan, & Alles, 2002).

The aforementioned scandals lead to the drafting and passing of legislation, such as the Sarbanes-Oxley Act of 2002 in the United States, as an attempt to restore investor confidence. Section 409 of the Sarbanes-Oxley Act, which relates to financial reporting, requires reports on a 'rapid and current basis' (Alles et al., 2004). Compliance to section 404 of the Act, which relates to providing assurances of controls, also endorses the adoption of continuous auditing (Srinivas, 2006a). It is suggested that the demand for continuous auditing, created by regulations, is a critical driver of continuous auditing (Alles et al., 2004). The need which is highlighted, is the need for reliable, high-quality information on which to base decisions.

2.4 Conclusion

This chapter examined what continuous auditing is by discussing the definitions offered in literature. It was established that continuous auditing is useful to both

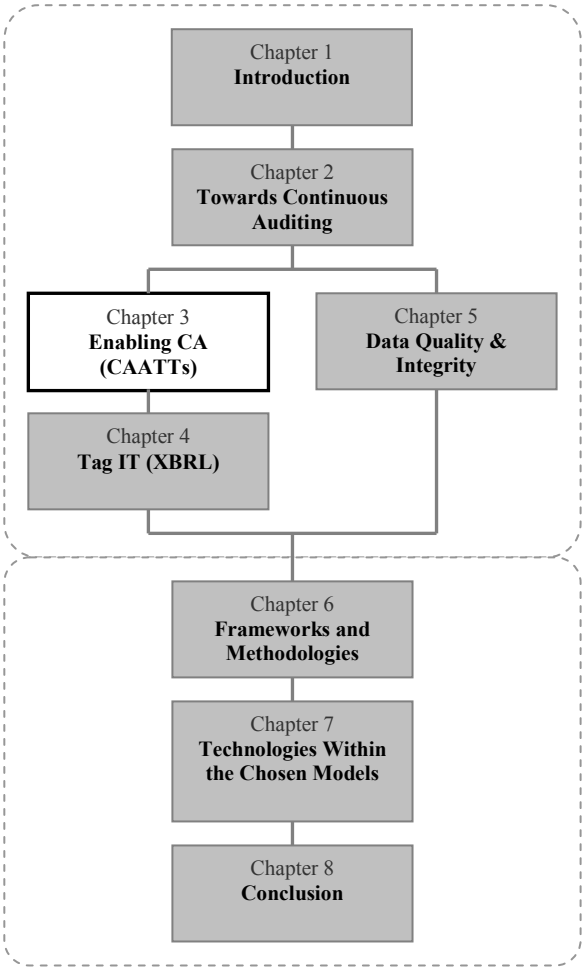
internal and external auditors. Although, it is the internal auditors who are the most effective at implementing continuous auditing techniques (because of their knowledge and access to systems), external auditors also benefit from continuous auditing, as they rely on the work of internal auditors as well as their own knowledge and expertise (Srinivas, 2006a).

It was also shown that the desire for continuous assurances, as provided by continuous auditing, are bound up in a complex relationship with the technologies which support continuous auditing. At the time when information systems and information technology matured to a point where continuous auditing was feasible, several lapses in corporate governance resulted in the establishment of regulations such as the Sarbanes-Oxley Act, which brought about a need for instant and continuous assurances.

A major role of continuous auditing is to establish the integrity, quality and reliability of information on which reports are based. These reports provide information which is essential to making good decisions. The ability to make high-quality and timely decisions depends in part on the quality of the data and the existence of on-line and real-time information (Rezaee et al., 2001, p. 150). To this end, data quality is defined and discussed in Chapter Five and technologies which facilitate reliable real-time information exchange (XBRL) are discussed in Chapter Four. Before then, Chapter Three continues to look at the essential roles of technology in the development of continuous auditing systems, and introduces many audit tools and techniques which are aided by the use of computers.

Chapter 3

Enabling Continuous Auditing



Chapter 3:

3.1 Introduction

3.2 Advances in Technology

3.3 CAATTS

3.3.1 Analysing Transactions

3.3.2 Testing Internal Controls and Assessing Risk

3.3.3 Descriptions of Commonly Used CAATTs

3.4 Conclusion

3.1 Introduction

This background chapter, along with the next chapter, introduces the technological backdrop against which continuous auditing plays out. Continuous auditing was not always technologically feasible, but became increasingly possible due rapid advances in information technology, specifically during the 1990s. These technologies, which made continuous auditing possible, will be introduced. After which, the roles of software, known to accountants and auditors as computer aided tools and techniques (CATTs), will be discussed. This chapter will also look at the evolution of these tools and techniques into audit tools and techniques which are useful in the modern continuous auditing environment.

3.2 Advances in Technology

In the 1990s, some technical hurdles preventing the effective use of continuous auditing were perceived, the only imaginable way of overcoming these, was if three conditions were met (Shields, 1998). The first condition is that the information to be audited would have to be produced by a reliable system. The second is that the continuous auditing process would have to be highly automated. To this end, tools would have to be integrated into the client's system. The third and final condition is that there is a fast, accurate and secure communication channel available, for communication between the auditor's and the client's systems (Shields, 1998).

Technology became far more affordable in the last two decades and the capabilities of technology also increased. Srinivas (2006a) mentions five technologies which are required for continuous auditing. These are:

- ***More powerful processors***

  In order to perform real-time processing of transactions, a high level of processing power is required.

- ***Disk Mirroring* – RAID**

  Technologies such as RAID (redundant array of independent disks) have allowed more reliable mass storage of data to become possible.

- ***Vast amounts of cheap storage* – petabytes**

  As it may be desirable for continuous auditing to examine every transaction processed, large amounts of data storage may be required. Added to this, many auditing solutions require large databases and data marts. Archived data may also need to be stored for protracted periods, for future reference.

- ***Faster communication***

  One of the most important drivers is faster information exchange. Real-time reporting is not possible if the required information cannot be efficiently and promptly accessed. Increased network bandwidth and specifically the ability to communicate over extended networks, such as the Internet, are examples of this.

- ***Secure systems***

  Stored, transmitted and processed data, especially financial or performance information may be sensitive. The strong encryption algorithms which now exist are useful for securing data. When collecting digital evidence it is also essential that the evidence cannot be tampered with. Security also affects reliability, as discussed in Chapter Five, and continuous auditing requires a reliable system (Srinivas, 2006a).

According to Vasarhelyi (2002) the same technologies, which can create business threats, can also be used to facilitate, manage and assure business processes. Internetworking, digitalization of transactions, intelligent agents and improved analytics are mentioned as examples of these technologies.

3.3 CAATTs

It can be established that, in order to verify that a real-time accounting system is producing reliable and credible financial information, testing of controls must be done simultaneously with substantive tests of transactions (Helms & Mancino, 1999; Rezaee et al., 2001). There are various tools and techniques which aid in analysis of transactions and internal controls. These tools are required to perform a variety of tasks, and can either be purchased software packages or auditor-designed routines (Rezaee et al., 2001). Collectively these tools and techniques are often referred to as Computer Aided Tools and Techniques or CATTs. An alternative acronym is CAATTs or Computer Aided Audit Tools and Techniques. CATTs have been used by auditors for many years and incorporate a wide variety of technologies, some of which are applicable to continuous auditing. In certain literature these have become known as Continuous Auditing Tools and Techniques. In this paper 'CAATTs' will refer to all of these collectively.

CAATTs in the broadest sense, may be viewed as, any technology which assists the auditor in completing his/her audit duties, for example, working papers and word processors (Braun & Davis, 2003). Software suggested to be CAATTs would therefore include: word processing, text search and retrieval tools, reference libraries, spreadsheets, presentation tools, utility software, flowcharting software, software

licensing checkers, electronic questionnaires, control self assessment, data warehouses, expert systems, data mining applications, and a variety of audit management, administration and security analysis software (Coderre, 2001). However, the term is often used by auditors to refer only to those tools and techniques which extract and analyse data during an audit (Braun et al., 2003).

According to the South African Institute of Chartered Accountants (SAICA) handbook, CAATTs may be categorised as package programs, purpose-written programs, utility programs or system management programs (South African Institute of Chartered Accountants, 2003).

- Package programs are usually generalised computer programs used to perform data processing functions. These include, reading data, selecting and analysing information, performing calculations, creating data files and producing reports. Examples include GAS tools such as ACL and IDEA.

- Purpose-written programs are used to perform audit tasks when specific circumstances occur. These may be written by the auditor or by a programmer instructed by the auditor. Normally the applications of the organisation being audited are used (they may be modified), because it is more efficient than developing independent audit software.

- Utility programs perform data processing functions such as sorting, creating and printing files. These programs may lack features such as automatic record counts or control totals, as they are not intended specifically for audit use, e.g. Microsoft Excel and Access.

- System management programs are usually part of the operating system and include data-retrieval software or code-comparison software. As with utility programs, these programs are also not designed for auditing use, e.g. AS/400 data management console

- Embedded audit routines, also known as embedded audit modules, are built into the audited entity's computer system in order to gather data on behalf of

the auditor. The two most common methods of using EAMs are the snapshot approach and System Control Audit Review File (SCARF) (Rezaee et al., 2001). The snapshot approach involves taking a 'picture' of a transaction as it is processed by an application. Embedded audit routines continuously capture images of the transaction, throughout the processing stages. This allows the auditor to see the progress of data through the system and evaluate the processes applied to the transaction. SCARF entails assimilating data regarding transactions, collected by embedded audit modules, into a special file. This file can then be examined by auditors. This provides the auditor with a method to continuously monitor transactions.

- Test data techniques allow the auditor to enter data into the entity's system and compare the results to a set of expected results for that data.

Further, according to the SAICA Handbook, CAATTs can also be used in many auditing procedures including:

- Tests of details of transactions
- Analytical procedures
- Tests of general controls
- Sampling programs
- Tests of application controls
- Re-performing calculations done by an accounting system (tests of balances)

For the purpose of this chapter, these can be grouped into two broader categories. Firstly CAATTs use for *analysing transactions*, and secondly those used for *testing internal controls and assessing risk*. Testing the details of transactions, analytical procedures and re-performing accounting calculations all relate to analyzing transactions. Tests of general controls and tests of application controls relate to testing controls. Sampling programs, which extract data, are possibly useful in either case.

CAATTs are often used in both of the broader activities mentioned above, but before examining in detail, these tools and techniques, it is necessary to explore why they are useful for these activities. One must bear in mind that both these audit activities are

usually performed concurrently, and while certain types of CAATTs are more suited to certain tasks, some may be applicable to both activities.

3.3.1 Analysing Transactions

To meet the requirements of an audit it is necessary to verify the accuracy of transactions in order to reveal fraud or error. Substantive tests of transactions must be performed. These will aim to obtain evidence showing possible material misstatements in the financial statements (South African Institute of Chartered Accountants, 2003). Two types of substantive tests are performed: *analytical procedures*, and *tests of transactions and balances* (South African Institute of Chartered Accountants, 2003):

1. Analytical procedures involve performing comparisons of financial data to establish a relationship. Often ratios are calculated. Besides easily indicating the existence of possible financial misstatements, analytical procedures can help reveal to the auditor the way the client's industry and business functions. When performed in the final phase of an audit, analytical procedures allow the auditor to comment on the reasonableness of transactions and the ability of the client to continue as a going concern. CAATTs make analytical procedures more feasible and affordable than before (Rezaee et al., 2001). Many types of analytical procedures are too complex or time-consuming to be done manually. Using CAATTs also means that it has become possible to use larger sets of data when performing analytical procedures.

2. Transactions are tested continuously, throughout the financial year. This helps to reduce the number and or complexity of tests of balances which need to be performed after balance sheet date (Rezaee et al., 2001). This is done to discover whether material misstatements have occurred; whether erroneous or irregular processing of the transactions has taken place. Tests done on balances are usually to collect evidence, on which the auditor can ground his or her opinion on fair representation of financial statements (Rezaee et al., 2002). When performing substantive tests of balances, Generalized Audit Software tools are often used (Rezaee et al., 2001).

The CAATTs most suited to analyzing transactions are Generalized Audit Software (GAS), Embedded Audit Modules (EAMs) and Artificial Intelligence related technologies.

3.3.2 Testing Internal Controls and Assessing Risk

In order to plan an audit the auditor needs to be aware of the areas which carry the greatest risk and thus need the most scrutiny. This requires the auditor to look at the adequacy and effectiveness of internal controls within the system. According to the Statement on Auditing Standards No. 80 (American Institute of Certified Public Accountants (AICPA), 1996) CAATTs can be used for this purpose. Testing of controls should also be ongoing. This allows the auditor to express an opinion as to how reliable the internal control system is. Knowing the reliability of the internal control system is important during the planning phase of an audit, as this determines the nature, timing and extent of substantive tests (Rezaee et al., 2001).

CAATTs which can be used for testing internal controls include: using test data, integrated test facilities (ITFs), embedded audit modules (EAMs), parallel simulation and concurrent processing (Cerullo et al., 2003). Neural networks may also be of some use.

3.3.3 Description of Commonly Used CAATTs

Now that the purposes of CAATTs, in the two main audit activities have been explained, some of the commonly used CAATTs can be described:

- ***generalized audit software (GAS)***

  According to Braun and Davis (2003), *GAS is the most frequently used of all CAATTs*. Widespread use may be due to a few reasons. GAS tools are simple to use (especially in comparison to other CAATTs). They require the auditor to possess minimal information systems knowledge. GAS tools are relatively easy to customize, and can be adapted for use in many different types of systems. GAS tools cause minimal disruption to the client's systems and do not need a high level of reliance on the client.

GAS can be used to perform many functions. Mostly, these functions are related to analyzing data, extracting information from the client's systems and aiding the auditor in his/her daily operations. Examples of GAS tools' uses include: footing ledgers, counting records, stratifying accounts by size, extracting data, downloading information for analytical review, selecting samples for detailed audit testing, generating confirmations, and identifying and reporting exceptions and unusual transactions (Glover and Romney, 1998; Lanza, 1998 in (Bierstaker et al., 2001). GAS tools are mostly used for substantive testing but can be used for limited testing of controls (Cerullo et al., 2003).

The two most popular commercially available GAS tools are ACL (Audit Command Language) and IDEA (Interactive Data Extraction & Analysis) (Bierstaker et al., 2001; McCollum & Salierno, 2003). Other commercial, package programs, include: Audicon, Autoaudit, Auditserve, Rapport and Pentasafe (Onions, 2003). Many auditors would class Microsoft Excel and other spreadsheet applications as simple GAS tools. In a survey, 51% of the internal auditor respondents still used general-purpose applications, such as Microsoft Excel and Access, instead of commercial GAS tools (Ramamoorti et al., 2004).

- ***Artificial Intelligence (AI) and related technologies***

Various types of AI software can be used in auditing, including: autonomous agents, expert systems and neural networks.

There are several varieties of autonomous agents; they may be called control agents, digital agents, autobots, softbots, webbots and robots (Debreceny & Gray, 2001). According to a definition by Franklin and Graesser (1996), "An Autonomous Agent is a system situated within and as a part of an environment that senses that environment and acts in it, over time, in pursuit of its own agenda and so to affect what it senses in the future". Agents differ from general software in that they are reactive and can also be proactive (Kogan, Nelson, Srivastava, Vasarhelyi, & Bovee, 1998). According to Woodroof and Searcy (2001) in the context of continuous auditing; "a digital agent is a set of electronic instructions (software) that acts [sic] on behalf of the auditor in a semi-autonomous manner to perform some service related to the subject matter being audited". Control agents

are a subset of agents specifically tailored for auditing. They are described as audit programs which use a set of auditor-defined heuristics to analyze a transaction set. This proactive software looks for patterns of activity similar to suspect unusual activities, using technologies such as digital analysis and data mining. If no explanation is found for the unusual activities detected, it then alerts the auditor to the presence of unusual activity (Kogan, Sudit, & Vasahelyi, 2000).

Many authors have suggested various uses for agents in auditing. They can be used to gather or sort information, but can also perform analyses and make decisions related to financial data (Debreceny et al., 2001). FRAANK (Financial Reporting and Auditing Agent with Net Knowledge) is an example of an audit agent. FRAANK is designed to extract data from *natural text* financial statements and translate them into XBRL statements (Kogan et al., 1998). This could be used in numerous ways in a continuous auditing system. For example when new statements produced in XBRL need to be compared to historical statements, which may be in a legacy system format.

In real-time accounting systems, agents can replace confirmations of receivables or payables, by performing continuous queries into third party systems. This makes the system more sensitive to fluctuations than traditional systems and allows for Continuous Online Assurances. They can show discrepancies by confirming receivables and payables and reconciling cut-off and float differences (Vasarhelyi, 2002).

Another important AI technology is Expert Systems. Expert Systems are able to process huge amounts of data in an intelligent way, mimicking human analysis. They interpret data to find patterns. While traditional audit procedures look for specific anomalies in known data patterns, expert systems have the advantage of being able to determine patterns where no previous patterns are known to exist (Dalal, 1999). Expert systems can be advantageous in analysing huge amounts of historical data, a task which would be too time-consuming and expensive for a human to perform.

One of the latest AI related technologies to garner attention is Neural Networks. Neural Networks process data in a way similar to the human brain. Like Expert Systems, Neural Networks can analyze huge sets of data. In fact, because they learn by trial and error, larger data sets improve the accuracy of the neural network over time. They work by recognizing new patterns within the data. Neural Networks can be used in many areas including; risk assessment and assessing internal controls. They can also be applied as a forensic accounting tool to proactively predict the occurrence of fraud (Cerullo & Cerullo, 2006). While neural networks are particularly suited to fraud detection, they are also useful in risk assessment and testing internal controls (as well as other business tasks such as visualising complex databases for marketing segmentation). Neural Networks are advantageous in that they can derive meaning from imprecise data and are capable of recognising trends which are too subtle or complex for humans to notice (Koskivaara, 2003). They are suited to solving problems ranging from simple to complex, in addition to, structured and unstructured problems. Thus, neural networks can solve a broader range of problems than technologies such as Expert Systems, as they can solve problems which appear nearly random in nature. The major advantages are adaptive learning, self-organisation, real-time operation and a high fault tolerance (Cerullo et al., 2006)

AI and related technologies are becoming more commonplace in everyday life and will soon be essential tools in real-time accounting and auditing systems. Intelligent agents for example are able to complement, and sometimes replace, the functionality provided by embedded audit modules (Vasarhelyi, 2002).

- ***embedded audit modules (EAMs)***

As long ago as 1989, embedded audit modules were proposed as a method of capturing audit information on a continuous basis (Groomer & Murthy, 1989). In fact, embedded audit modules, along with integrated test facilities (ITFs), can be called *continuous* audit approaches (Cerullo et al., 2003). This is because they are applicable to systems performing real-time financial reporting on transaction data.

EAMs are audit related routines within the source code of the application. They are designed to continuously monitor events which are significant to the audit and

then report on them. They do this by identifying and flagging transactions which meet pre-set criteria. These transactions are then reviewed by the auditor and this can be done in real-time or in batch (Braun et al., 2003).

EAMs are highly automated, and function with little intervention. They are used by auditors in two main ways. Firstly, when testing transactions, EAMs can be used to identify large numbers of transactions for substantive testing. Secondly, EAMs can be used in the evaluation of control risk. They test controls by checking if transactions are processed according to procedures and policies (Cerullo et al., 2003).

Some negative aspects of EAMs are related to the fact that they are built into the client's application. Not only does this mean that a substantial amount of planning is required in designing an application, but also that programming expertise is required to implement and maintain the module. Changes to the application could also require the EAM code to be reviewed. Auditors would need to have a close relationship with the client's system administrators (Braun et al., 2003).

- ***integrated test facilities (ITFs)***

As mentioned above, ITFs are useful in a real-time accounting environment. The main difference between EAMs and ITFs is that EAMs examine data processed by the application to infer the quality of the processes in that application, while ITFs directly examine the internal logic of the client's application (Braun et al., 2003). This makes ITFs effective for evaluating application controls. To do this, code modules are built into applications. These are built to discriminate "dummy data" from "live" data. The test, "dummy" data run through the normal data stream, and the results of the dummy data are used by the auditor to evaluate application controls during normal operations (Braun et al., 2003).

There are, however, some disadvantages to using ITFs. There is a higher risk of corrupting data than some other CAATTs. Extra controls need to be in place to ensure the effects of dummy data are removed. ITFs also often require a high level of computer programming expertise to create and maintain (Braun et al.,

2003). The implementation of an ITF can therefore become time-consuming and costly (Cerullo et al., 2003).

- ***noncontinuous audit techniques***

Tools such as Parallel Simulation, Concurrent Processing and Continuous and Intermittent Simulation are also used to evaluate internal controls. However, both Parallel Simulation and the use of Test Data are described by Cerullo and Cerullo (2003) as *noncontinuous audit techniques.* They are not suited for use in continuous auditing, and are more applicable to periodic financial reporting, as they cannot be performed in real-time.

Parallel Simulation involves taking actual transaction data from the client's system and re-entering it into another system. This system may be an audit software package or an accounting/ERP package such as SAP R/3, PeopleSoft, BusinessWorks or Oracle Financials. The output from the duplicate system is compared to that produced by the client's production system. The results should ideally be the same; differences may reveal problems within the client's system (Cerullo et al., 2003). This allows the auditor to analyse the quality of the process performed by the client's application (Braun et al., 2003).

3.4 Conclusion

This chapter firstly introduced some technological advances which made continuous auditing more feasible, and then introduced CAATTs. Section 3.2 introduced the three conditions which needed to be met in order for continuous auditing to become technologically feasible. Rapid developments in technology, particularly during the 1990s meant that these became fulfilled. Increased systems integration, possibly due to the use of ERP systems, and the ability to share common data aided in fulfilling the need for reliable systems. Advances in communication technology and encryption met the need for fast, accurate and secure communication. The final requirement was for a highly automated audit process. This is where CAATTs, such as GAS and EAMs, played a major role (Shields, 1998).

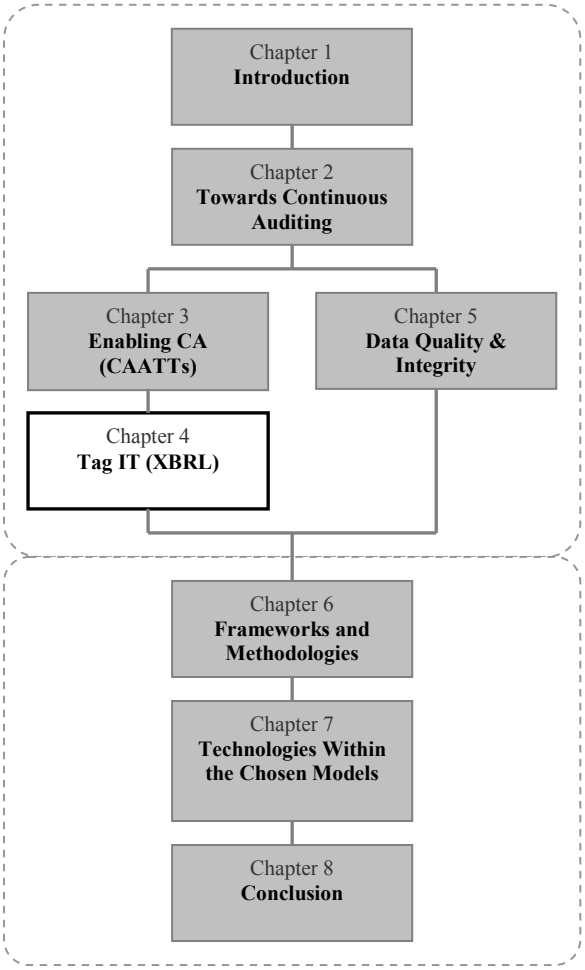
The next section explored what CAATTs are, and how they can be used in auditing. It was established that technologies, such as GAS, can assist the auditor in his/her

duties. It is important to remember that merely using computer software in auditing is not enough, as the audit process has evolved from manual auditing of accounting systems (which have paper documentation) to an online, continuous audit of paperless, electronic EDI systems (Rezaee et al., 2001). With this in mind, it was established that some CAATTs are useful in continuous auditing, particularly, expert systems, agents, embedded audit modules and integrated test facilities.

These tools and techniques will be placed in context within continuous audit systems in Chapters Six and Seven. In the next chapter, the development of information technology to support continuous auditing will be further discussed. In particular, the roles of XBRL in continuous auditing will be clarified.

Chapter 4

Tag IT



Chapter 4:

4.1 Introduction

In the previous chapter, the technologies which enable continuous auditing were examined. Another technology, which promises to facilitate the adoption of real-time reporting principles, is eXtensible Business Reporting Language (XBRL). In this chapter, XBRL will be defined, and some of its inner-workings will be explained. The general advantages, as well as some specific benefits to continuous auditing will be discussed. After which, the limitations will be explored. The last section focuses on technologies similar to XBRL. Finally, the chapter will be concluded with an explanation of the importance of XBRL to continuous auditing.

XBRL is a freely available, open specification for financial reporting. It was developed by a consortium established in the second half of 1999. This consortium was a joint effort between industry and the American government, and it included the American Institute of Certified Public Accountants (AICPA), information technology companies and some of the largest accounting and professional services firms[1] (Coderre, 2004; Debreceny et al., 2001).

XBRL is designed to assist financial professionals to extract relevant data and to prepare, publish, exchange and analyze business reports, specifically, financial statements (Coderre, 2004; Hannon, 2005). Both humans and intelligent agents would be able to use information, distributed via the Web, with improved accuracy and reliability (Debreceny et al., 2001). It allows reports to be published in a variety of formats, such as HTML, XHTML, PDF and spreadsheets, such as Excel (The Canadian Institute for Chartered Accountants, 2002). Although it is a standardised language, XBRL is also extensible, so it can be tailored to meet business requirements.

XBRL aims to facilitate easier decision-making, by making the information more reliable. However, it does not guarantee the correctness and completeness of data for making informed decisions (Srinivas, 2004). This will be discussed in more detail in the next section. Although, it is also argued that XBRL can help to improve the data quality within the system[2] (Coderre, 2004; Willis, 2005; Willis & Hannon, 2005), in addition, according to the webpage of the XML steering committee, http://www.XBRL.org, 'it provides major benefits in the preparation, analysis and communication of business information. It offers cost savings, greater efficiency and improved accuracy and reliability to all those involved in supplying or using financial data.'[3]

These are some of the reasons for the existence and desirability of XBRL. How XBRL works will be clarified in the following section.

[1] More detailed information, including current members of the XBRL steering committee, is available at www.XBRL.org
[2] Data Quality is discussed in detail in the next chapter, Chapter Five.
[3] http://www. xbrl.org/frontend.aspx?clk=LK&val=20 (Introduction to XBRL)

4.2 How XBRL Works

XBRL is built on XML (Extensible Markup Language). XML is a standard used for data exchange on the Internet. XML uses tags to identify items of data. Each tag consists of metadata included between '<' and '>'. An un-nested pair of tags, opening and closing, describe the data encapsulated between them. XML is known as 'meta-language'. This refers to the fact that XML describes how to write a language, and it is not merely a language in itself. Consider, as an analogy, that it is a grammatical rule that a sentence begins with a capital letter and ends with a punctuation mark, usually a full-stop. This is a simple 'meta' description of a sentence. XML tags describe data in the same way. The sentence itself is portrayed by the grammar of the language; in the same way the actual data is carried, enclosed by tags (Garthwaite, 2000). This allows for software to efficiently and easily process the data.

Unlike HTML[4], XML does not say how data must be presented. XBRL, like its parent technology, XML, is not designed to render information for human viewing and use. Instead, it is designed to enhance the efficiency of data transfer and archiving (Boritz & No, 2004). An example of an XBRL document is as follows (http://www.XBRL.org):

```
<ifrs-gp:AssetsHeldSale contextRef="Current_AsOf" unitRef="U-Euros"
  decimals="0">100000</ifrs-gp:AssetsHeldSale>
<ifrs-gp:ConstructionProgressCurrent contextRef="Current_AsOf"
  unitRef="U-Euros" decimals="0">100000</ifrs-
  gp:ConstructionProgressCurrent>
<ifrs-gp:Inventories contextRef="Current_AsOf" unitRef="U-Euros"
  decimals="0">100000</ifrs-gp:Inventories>
<ifrs-gp:OtherFinancialAssetsCurrent contextRef="Current_AsOf"
  unitRef="U-Euros" decimals="0">100000</ifrs-
  gp:OtherFinancialAssetsCurrent>
<ifrs-gp:HedgingInstrumentsCurrentAsset contextRef="Current_AsOf"
  unitRef="U-Euros" decimals="0">100000</ifrs-
  gp:HedgingInstrumentsCurrentAsset>
<ifrs-gp:CurrentTaxReceivables contextRef="Current_AsOf" unitRef="U-
  Euros" decimals="0">100000</ifrs-gp:CurrentTaxReceivables>
<ifrs-gp:TradeOtherReceivablesNetCurrent contextRef="Current_AsOf"
  unitRef="U-Euros" decimals="0">100000</ifrs-
  gp:TradeOtherReceivablesNetCurrent>
<ifrs-gp:PrepaymentsCurrent contextRef="Current_AsOf" unitRef="U-Euros"
  decimals="0">100000</ifrs-gp:PrepaymentsCurrent>
<ifrs-gp:CashCashEquivalents contextRef="Current_AsOf" unitRef="U-
  Euros" decimals="0">100000</ifrs-gp:CashCashEquivalents>
<ifrs-gp:OtherAssetsCurrent contextRef="Current_AsOf" unitRef="U-Euros"
  decimals="0">100000</ifrs-gp:OtherAssetsCurrent>
<ifrs-gp:AssetsCurrentTotal contextRef="Current_AsOf" unitRef="U-Euros"
  decimals="0">1000000</ifrs-gp:AssetsCurrentTotal>
```

[4] Html (Hypertext Markup Language) is the standard used for presenting Webpages on the Internet

While XML-based languages make data processing by software easier, to make the information carried within the tags readable for humans, however, the addition of a style sheet is required. The style sheets add the necessary presentation elements to the data in XBRL documents, and the results can then be presented in HTML, PDF or another preferred presentation format. Style sheets may be Cascading Style Sheet (CSS) files, Extensible Style Sheets (XSL), spreadsheets or another technology, which is used to produce the reports (The Canadian Institute for Chartered Accountants, 2002). The HTML output, produced by rendering the example code with a style sheet, is shown below (http://www.XBRL.org):

CURRENT ASSETS	
Assets Held for Sale	100,000
Construction in Progress, Current	100,000
Inventories	100,000
Construction in Progress, Current	100,000
Hedging Instruments, Current [Asset]	100,000
Current Tax Receivables	100,000
Trade and Other Receivables, Net, Current	100,000
Prepayments, Current	100,000
Cash and Cash Equivalents	100,000
Other Assets, Current	100,000
Current assets, Total	1,000,000

XBRL, as a standard, consists of a few components. These include the specification and taxonomies, as well as (the previously mentioned) instance documents and style sheets.

The XBRL Specifications are the technical documents, which outline what XBRL is, and how it works. It is the framework of XBRL, and explains the syntax and semantics of XBRL taxonomies and instance documents (Srinivas, 2004). It is the foundation of all XBRL reporting, as it contains all the rules used to express business report information in compliance with the format specified by the World Wide Web Consortium (W3C) (Hannon, 2005).

The two essential components of XBRL are instance documents and taxonomies. Instance documents contain the actual data or facts, while taxonomies are the lists of tags which have been agreed upon for expressing financial concepts (Hannon, 2005).

Taxonomies are analogous to a dictionary, which describes the words of a language, and how they are used (http://xbrl.org/frontend.aspx?clk=SLK7val=37). Taxonomies detail the concepts and interrelationships used for a particular type of business reporting, which is usually in the form of a set of rules, such as Generally Accepted Accounting Principles (GAAP) (The Canadian Institute for Chartered Accountants, 2002). Taxonomies help to ensure that the same tags are used in the same way by all publishers of XBRL documents (who use the same taxonomy). This allows comparison between the financial reports of different companies. Examples of taxonomies include: US GAAP CI, XBRL GL (General Ledger) and the IAS's (International Accounting Standards) GAAP CI taxonomies. There are also several national taxonomies, which are necessary because of differences in regulations and reporting requirements from country to country. These include Canadian, German, Australian, New Zealand and Singapore GAAP CIs. Since XBRL is based on XML, it is extensible. This means that if a taxonomy does not contain a tag suitable for the user's needs, a custom tag can be created. In effect, a customized version of a taxonomy can be created (Boritz et al., 2004). Taxonomies are composed of two parts. The first part is one or more schema documents. These contain a declaration (list) of elements that can be used in an instance document. The second is linkbase files, which are used to describe additional information about the elements in schema documents. This includes the relationships between elements in schema documents. Five types of linkbases exist: Label, Definition, Reference, Presentation and Calculation (Hannon, 2005).

As previously stated, an instance document is a collection of data elements that are tagged in accordance with the taxonomy being used. The relationship between the taxonomy and the instance document is that while the taxonomy describes the items as elements, instance documents hold the actual amounts or details of the items (The Canadian Institute for Chartered Accountants, 2002). The relationship between these components is clearly illustrated by the following example.

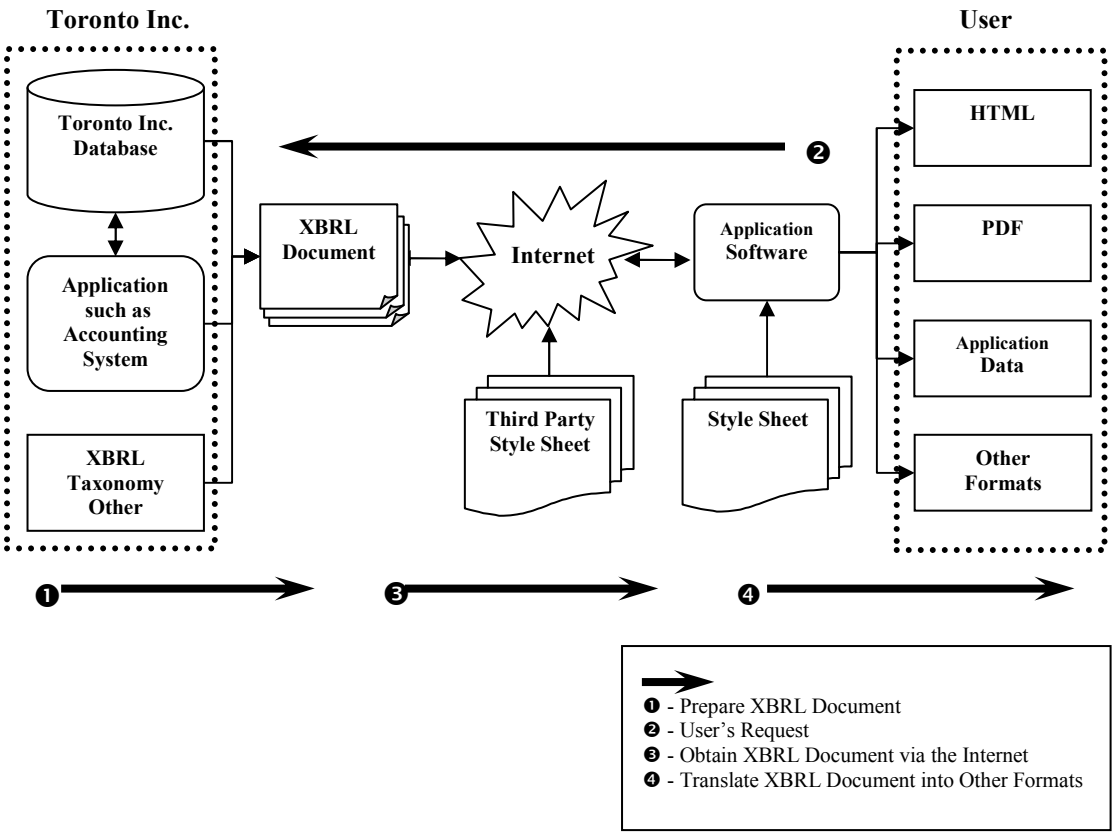


Figure 4.1: How XBRL Works (Boritz et al., 2004)

In the diagram, an example of XBRL use is given for a fictitious public company named 'Toronto Inc'. Firstly, in order to make the financial statements available to analysts, the financial information must be prepared. This is usually done using the company's accounting package. A XBRL document is created and validated. Validation is checking if it is a well-formed XBRL. This document is then published on the company's Website or FTP server. The user (analyst) can then request the necessary information, returned in XBRL format, via the Internet. The user could make this request through a Web application interface. Since XBRL is not a format that is clear to the human reader, the document may be parsed and transformed into a suitable format, before being delivered to the user. This transformation is achieved through the use of style sheets, which may be part of the requesting user's application, or it may be a third party style sheet, made commonly available on the Internet. The user's application may alternatively import the XBRL data into another system for further processing (Boritz et al., 2004). This could allow the analyst to create simulations of the system, and may be particularly useful during an audit. This illustrates one of the benefits, which are discussed in the next section.

4.2.1 Benefits of XBRL

XBRL is seen to be beneficial. Analysts, investors and regulatory bodies gain faster and more efficient access to data regarding public companies from its use. This information would also be in a format which will facilitate the production of valuable information for decision-making. An every-day example, which benefits from XBRL, is loan approval. If a loan request is submitted to a bank in XBRL format, it can immediately be inputted into an analysis tool. A decision as to whether the loan is accepted or rejected, can be made in minutes. In contrast, using EDI and legacy systems; this same task may take days. This is due to the time-consuming nature of entering and re-entering the required data into each system (Pinkster, 2003). This is made unnecessary when using XBRL.

Another of the perceived benefits of XBRL is that it improves data quality (Coderre, 2004; Willis, 2005; Willis et al., 2005). The improvements in data quality are attributed to the automation of information exchanges between different software applications (Willis, 2005). Poor data quality is often due to data losing context during transmission, as data is not always validated independently when it is exchanged between software applications. According to Willis and Hannon (2005), this problem occurs most frequently where a centralized data warehouse is implemented, and where the data is consumed by many disparate data stores. XBRL aids in maintaining data quality because it allows automatic validation when data is exchanged between software applications (Willis et al., 2005). This is possible because XBRL can supply the logic behind data; it is able to inform the systems of what to expect when data is automatically re-entered (Cohen, 2002). The logic is a description of each data element - 'metadata'. XBRL minimizes the risk of losing metadata – this helps to maintain the accuracy of data (Naumann, 2004).

Efficiency of auditing is also improved, because XBRL provides a standardised data format (Naumann, 2004). XBRL can be used regardless of operating system or software application, and it allows multiple use and re-use of data as it is not in a proprietary format (Hannon, 2005; Pinkster, 2003). Reports and statements can be created once and then used in several ways (published on the web, printed as reports or submitted as regulatory filings) without duplication of effort. This also reduces the

need for data to be re-keyed when exchanged (Naumann, 2004). This increased efficiency and effectiveness can produce cost savings (Coderre, 2004).

XBRL improves transparency when auditing. This is because user access to information is greatly increased (Pinkster, 2003). The use of XBRL can aid in meeting the reporting obligations of the Sarbanes-Oxley Act of 2002, specifically section 409, which requires real-time reporting (Coderre, 2004; Naumann, 2004). Real-time reporting has also become desirable, regardless of regulations, because it facilitates better decision-making.

These benefits of XBRL may help to facilitate the use of continuous auditing. Amongst the requirements for effective continuous auditing, is the need for reliable systems and timely audit reports. XML-based reporting, such as with XBRL, along with XML Web Services, could facilitate more reliable systems (Cohen, 2002). XML-based standards also aid in increasing the timeliness of reports, due to increased efficiency in exchanging data between systems. XBRL, in particular, also facilitates the use of Web-enabled audit programs for standards-based financial statement reviews (Coderre, 2004). Web-enabled audit programs are often a key part of continuous auditing, as illustrated by the continuous auditing models, proposed by Woodroof and Searcy (2001) and Onions (2003), which are discussed in detail in Chapter Six. Perhaps the single biggest advantage of XBRL is that is creates a standardised data format for describing business information. According to Cohen (2002), because of the current focus on internal controls behind the financial reporting process, 'being able to create, and extract information from, financial statements on a machine-consistent, reliable and testable fashion is vital'.

Taxonomies, such as XBRL GL, and the Journal Taxonomy, may also hold advantages for the adoption of continuous auditing. As XBRL GL is an extensible tool, which uses a standardised format to represent the key information in business documents and transactions, it creates a 'universal audit trail'. XBRL GL represents the information usually found in the General Ledger, where most accounting programs store fields of extra information. The information may represent the chart of accounts' information, customer/vendor/employee master files, inventory history, stock status, open receivables and payables, as well as other accounting information.

XBRL GL can also be used to capture performance metrics, not merely a monetary amount, as performance metrics can be linked to XBRL identifiers. These features will allow automated systems to easily trace through financial reporting. The system could analyze summarized reports, verify the underlying details in the accounting system and then even link back to the source documents and electronic transactions. Transactional data is also available in any level of granularity. Continuous auditing tools can be designed to monitor XBRL GL details so that they can anticipate what should be reported on, and trigger further analysis when results vary from what is expected (Cohen, 2002).

In summary, some of the generally perceived benefits of XBRL, which make it desirable to practitioners, are that it improves accuracy, efficiency and transparency of the audit process (Coderre, 2004; Naumann, 2004). XBRL also facilitates continuous auditing because, as a standardised data format, it increases the efficiency of exchanging data between systems. Taxonomies, which represent the General Ledger (such as XBRL GL), also facilitate the creation of efficient continuous auditing systems as they create an audit trail, which can be accessed with varying degrees of granularity. However, XBRL also has some limitations, and these are explored in the next section.

4.2.2 Limitations and Problems with XBRL

XBRL was originally designed to describe financial statements; however, it has grown in purpose to describe business information. One limitation is that it cannot, using tags, convey graphs, formulas and charts. Some users may prefer multimedia data formats or multi-dimensional numerical and graphic presentations. These formats may not be adequately described in XBRL (Boritz et al., 2004).

A further problem is that XBRL can introduce security risks. These risks are mostly due to the online nature of XBRL. As it enables seamless communication along the financial supply-chain, end-to-end security is required for all XBRL communication. This will include authentication, access control, malicious attack prevention, non-repudiation and data privacy controls (Coderre, 2004). Security issues will be discussed further in Chapter Five.

XBRL automates financial data exchange, and it aims to improve the process of extracting data from a variety of compliant systems. XBRL does not, however, guarantee the *correctness* and *completeness* of this data (Srinivas, 2004). Boritz and No (2004) suggest that XBRL does not address the quality of information, for example, whether the data described by a XBRL instance document is reliable. Reliability may be compromised by applying the inappropriate taxonomy. Data would then be described by the incorrect tags. This may happen if procedures used for preparing financial statements are incomplete or inaccurate (Coderre, 2004). Added to this is the issue of trustworthiness of the data. Like most data published on the Web, the source can be disguised. Data may also be created or edited, without leaving a trace (Boritz et al., 2004).

The accountant's report is viewed as part of the financial statement, as a whole. This means that XBRL does not provide for an assurance report on an individual financial statement, an individual item within a statement or the internal controls which underlie the financial report. Boritz and No (2004) suggest a solution to this problem, as well as the issues regarding the reliability (specifically correctness and completeness) and trustworthiness of the data. The proposed solution is an XML related technology, named XARL (Extensible Assurance Reporting Language), which will be discussed in the next section (Boritz & No, 2003).

4.3 Related Tagging Technologies

There are many XML-related technologies in use. Some of these, relating to financial information, are explained later in this section. XARL, unlike the other technologies to be discussed, is designed specifically to work with XBRL. In fact, XARL is designed as an XML-based extension to XBRL.

XARL aims to contribute to the reliability of XBRL information. This is achieved using security techniques, such as Public Key Infrastructure (PKI) and XML security features. A XARL document includes tags explaining, for example, assurance type, assurance date, auditor's digital signature and system reliability. These tags would then allow users to identify the type of assurance, the time period covered in the

report, the assuror's name and the source of the assurance standards applied (which taxonomy was used).

4.3.1 XARL

The possible method of XARL implementation involves the usual production of XBRL documents by the company's accounting system. These are then sent, securely, to an assurance company (assuror). The assuror checks the validity of the XBRL and also performs assurance procedures. These procedures may include: confirming the reliability of the systems which produced the XBRL document, analytic procedures and obtaining evidence to support the amounts within the XBRL document. The assurance company then creates a XARL document, by mapping the assurance-related information into the elements, described by the XARL taxonomy. The assurance-related information may relate to the company's financial statements as a whole, individual financial statements or individual items in the financial statements. Assurance information regarding the company's information systems and controls may also be included (Boritz et al., 2003).

Users desiring information regarding the company can obtain the XARL document from the assuror. This may be either an on-line or off-line process. The documents are not made available on the Internet, as they are then vulnerable to interception, altering and spoofing. Users would need to provide the assurance company with their public key.

The assuror's XARL Authentication System would process the request and send the XARL document to the user, if authorized. This XARL document is digitally signed by the assuror, and encrypted with the user's public key. The identity of both the assuror and the user is authenticated by this process, and it also serves to record the exchange of information.

The user can then decrypt the document with their private key and the assuror's public key. This XBRL document can then be transformed into a desired format, using a style-sheet or imported into an application.

It is evident how using XARL reduces uncertainty regarding the reliability of the financial information, as the XARL tags provide evidence that the information has been audited by a public accountant and that it has not been tampered with (Boritz et al., 2003). However, XARL still does not guarantee the integrity and authenticity of information so an additional security infrastructure is also required[5].

4.3.2 Other XML-Based Standards

Other XML-based standards exist, which also describe financial information. These standards include ebXML, FpML, RIXML, MDDL, FIX, FIMXL, IFX and OFX (Srinivas, 2004), and are described below:

- ***ebXML (Electronic Business using Extensible Markup Language)***

  ebMXL was started in 1999. It was an initiative of OASIS and the United Nations/ECE agency (CEFACT). ebXML consists of a modular suite of specifications, which allows companies to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes. It enables communication between companies of any size, so that they may conduct business, via the Internet, regardless of their global region (www.ebxml.org).

- ***FpML (Financial Products Markup Language)***

  FpML is an industry-standard protocol for exchanging data regarding complex financial products between applications. It is a meta-language, used as a protocol for sharing information regarding swaps, derivatives and structured products. It is the standard used for electronic dealing and processing of financial derivatives instruments (www.fpml.org).

- ***RIXML (Research Information Exchange Markup Language)***

  RIXML was established by a consortium of buy-side firms, sell-side firms and vendors. It is an open specification, freely available for use in investment research. RIXML provides the ability to tag research content, which may be in

[5] The security infrastructure will be discussed in Chapter Five

any form or media. The tags provide details, according to which, the aggregated data can be searched, sorted and filtered (www.rixml.org).

- ***MDDL (Market Data Definition Language)***

MDDL provides a common data dictionary regarding the fields required to describe financial instruments, corporate events affecting value and tradability, and market-related economic and industrial indicators. It aims to allow entities to exchange market data by standardising formats and definitions. This allows market-data to be exchanged between systems more efficiently (www.mddl.org).

- ***FIX (Financial Information Exchange)***

FIX is a series of messaging specifications for electronic communication of trade-related messages. It is an open and free specification, which was developed with the aim of providing a common, global language for the automated trading of financial instruments (www.fixprotocol.org).

- ***OFX (Open Financial Exchange)***

OFX is a freely available specification, which was created in 1997 by CheckFree, Intuit and Microsoft. It is used for the electronic exchange of financial data, via the Internet, between financial institutions, business and consumers. It supports a wide range of financial activities, including consumer and small business banking, consumer and small business bill payments, bill presentment and investments tracking (including stocks, bonds and mutual funds). OFX became XML compliant in 2000 (www.ofx.net).

While these standards are also XML-based meta-languages such as XBRL, they do however differ in function from XBRL. Each of the above standards addresses a specific aspect of financial transactions. They are transaction-oriented, while XBRL is reporting-oriented (Srinivas, 2004).

4.4 Conclusion

A technological evolution has occurred. This evolution includes increased storage ability and faster and cheaper communication. Technologies, such as XBRL, are

becoming essential to the auditing process in a real-time environment, as they enable better reliability.

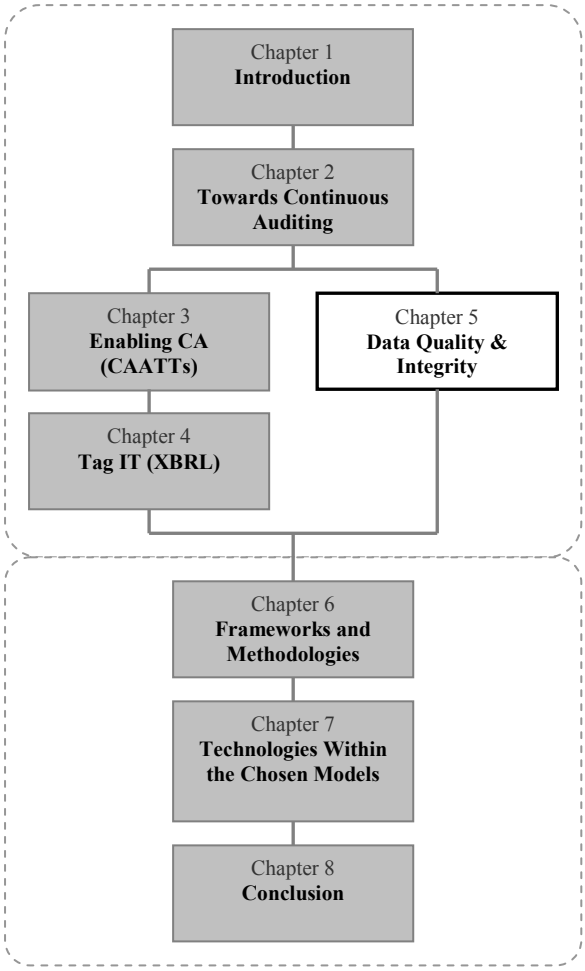
XBRL holds many advantages. As a standardised data format, it decreases the need for re-capturing data between systems, which is time-consuming and costly. It can be argued that this increases data quality, as it also conveys metadata along with the data. XBRL is also important in a real-time reporting environment, particularly in continuous auditing. Not only for the benefits it holds as a standardised data format, but it also facilitates the use of web-enabled audit programs, which are often integral in continuous auditing systems (Coderre, 2004).

As a new technology, XBRL also has certain shortfalls which were highlighted. Most notably it is not secure and does not inherently cater for correctness and completeness of data (data integrity). To this end XARL is suggested as an add-on to XBRL. Some of the many related XML-based technologies were introduced, and it was established that these differed from XBRL in that they are all transaction oriented, while XBRL is the only reporting language.

XBRL is set to become an important part of any continuous auditing solution, as a standardised data format. However XBRL is not a cure-all because of the myriad of legacy systems still in use. Many legacy systems are only capable of using data in proprietary formats and will not be directly aided by the adoption of XBRL. This issue will be addressed in subsequent chapters.

Chapter 5

Data Quality and Integrity



Chapter 5:

5.1 Introduction

The aim of continuous auditing, as defined by Rezaee et al (2002, p. 151), is to:

> …gather persuasive evidence regarding the quality and integrity of the system in producing reliable and credible information.

The attributes of quality and reliability of a system and the information within that system are highly important to continuous auditing. In this chapter, the relationship between continuous auditing and quality/integrity will be established, and the attributes of integrity and quality will be further explored.

The most simplistic definition of integrity is that it is the *'quality' or state of being whole, uncorrupted and complete* (Whitman & Mattord, 2003). The quality of information can be seen as a broader concept than integrity. Integrity is a prerequisite for information quality (Boritz, 2005).

Information quality refers to the relevance, reliability and usability of output data, whereas integrity refers to information possessing representational faithfulness, meaning that the information is complete, accurate/correct, current/timely and valid/authorized. The close relationship between quality and integrity is shown by the overlap of these components of integrity (representational faithfulness) with concepts related to quality, namely; relevance, reliability and usability (Boritz, 2004). As quality is a broader term than integrity, data quality will be discussed first.

5.2 What is Data Quality?

In literature, the terms *data* and *information* are often used interchangeably (Pipino, Lee, & Wang, 2002; Strong, Lee, & Wang, 1997). Therefore, unless otherwise specified, the term 'data quality' will be used to refer to both information and data quality. Data quality is not easily defined; but most often the commonly accepted dimensions of data quality are used in a definition (Wand & Wang, 1996). Boritz (2004) defines information quality simply as the relevance, reliability and usability of output data. But, according to Wand and Wang (1996), data quality is a multi-dimensional concept and its frequently mentioned dimensions are accuracy, completeness, consistency and timeliness[6]. A more comprehensive list of data-quality dimensions is given by Pipino, Yang and Wang (2002), who used them to develop data-quality metrics.

Dimensions	Definitions
Accessibility	The extent to which data is available, or easily and quickly retrievable
Appropriate Amount of Data	The extent to which the volume of data is appropriate for the task at hand
Believability	The extent to which data is regarded as true and credible
Completeness	The extent to which data is not missing and is of

[6] These dimensions/attributes overlap with the attributes of information integrity as discussed by Boritz (2004)

	sufficient breadth and depth for the task at hand
Concise Representation	The extent to which data is compactly represented
Consistent Representation	The extent to which data is presented in the same format
Ease of Manipulation	The extent to which data is easy to manipulate and apply to different tasks
Free-of-Error	The extent to which data is correct and reliable
Interpretability	The extent to which data is in appropriate languages, symbols, and units, and the definitions are clear
Objectivity	The extent to which data is unbiased, unprejudiced, and impartial
Relevancy	The extent to which data is applicable and helpful for the task at hand
Reputation	The extent to which data is highly regarded in terms of its source or content
Security	The extent to which access to data is restricted appropriately to maintain its security
Timeliness	The extent to which the data is sufficiently up-to-date for the task at hand
Understandability	The extent to which data is easily comprehended
Value-added	The extent to which data is beneficial and provides advantages from its use

Table 5.1: Data-Quality Dimensions (Pipino et al., 2002, p.212)

Data quality is important, because poor data quality drastically affects the effectiveness of an organisation (Wand et al., 1996). This is a concern, because data-quality problems affect organisational databases, and in turn, the information systems environment in which they function. This is particularly true when data is collected from multiple data sources (Strong et al., 1997). In continuous auditing systems, where information (processed data) is a primary product, having high-quality data should be a priority. High-quality data is that which is suitable for the needs of data consumers[7] (Strong et al., 1997). Conversely, data-quality problems are defined as problems affecting one or more quality dimensions, which makes the data completely or largely unfit for use (Strong et al., 1997). An area which can be negatively affected by data-quality problems is operations. For example, these problems can increase operational costs, as time and other resources are required to detect and prevent errors (Redman, 1998). Since continuous auditing systems aim to produce information in

[7] Strong et al (1997) define high-quality data as *"Data that is fit for use by data consumers"* p104.

real-time, delays due to poor data quality must be avoided. At the tactical level, poor data quality negatively affects decision-making (Redman, 1998). This is due to the fact that managers can often not reach a decision if they mistrust the data required to make it. Continuous auditing systems aim to help decision makers, thus the data which the continuous auditing system uses must be high-quality. Poor data quality further impacts the decision-making process by making the implementation of data warehouses very difficult (Redman, 1998). While the general use of data warehouses is to support decision-making, data warehouses perform the important function in continuous auditing of aggregating data, as will be shown in Chapter Six (Flowerday, Blundell, & R. von Solms, 2006).

According to Wand and Wang (1996), 'The quality of the data generated by an information system depends on the design of the system'. The later use of data is beyond a system-designer's control, therefore it is important to consider data quality, and the intended use of the data, when designing the system. Therefore, when designing a continuous auditing system, thought must be given as to how data quality will be provided for within that system. Since information integrity is a prerequisite for data quality, this too must be given consideration. In the next section the narrower concept of information integrity is discussed.

5.3 What is Data Integrity?

According to Boritz (2004), information integrity refers to information possessing representational faithfulness. This infers that information possesses the attributes of completeness, accuracy/correctness, currency/timeliness and validity/authorization. Boritz (2005) expands on the definitions used in COBIT (3rd edition) and CICA's Information Technology Control Guidelines (ITCG). COBIT uses the three attributes of completeness, accuracy and validity to define information integrity, while ITCG includes extra attributes, namely, authorization, timeliness, consistency and segregation of incompatible functions (Canadian Institute of Chartered Accountants, 1998). The definition as given by Boritz (2004), of information integrity having representation faithfulness, will be used in this dissertation.

In clearly defining information integrity, it is important to consider some related concepts. Two concepts, which are applicable to continuous auditing, are system

integrity and data integrity. System integrity is about whether the outputs fully and fairly represent the inputs of the system. It is important because it limits the highest attainable level of information integrity. This implies that the information integrity is reliant on the system integrity, that it cannot exceed the level of system integrity. However, the level of information integrity may be worse than that of system integrity due to certain factors which cause degradation, such as age of the information, errors, omissions, malicious acts and acts of nature (Boritz, 2004). The concept of data integrity is also important, as data is the raw material used to produce information. Data integrity is a narrower concept than information integrity.

Continuous auditing aims to provide information-integrity assurances on demand (Flowerday & R. von Solms, 2005b). In order to provide this, system integrity must be ensured, as the level of information integrity cannot exceed it (Boritz, 2004). But there is also a need to ensure data integrity, as data constitutes the raw material of information, so it is necessary to ensure its integrity in order to ensure information integrity.

5.3.1 How to Provide Integrity

One definition of data integrity states that it is the need to retain or preserve information (without alteration or corruption) from source to destination (Snedaker, 2006). This can be expanded upon using the NSTISSC Security Model, which shows that integrity must be maintained during data transit, storage and processing (National Security Telecommunications and Information Systems Committee, ; Whitman et al., 2003). This notion can be further expanded to all the processing phases of data suggested by Boritz (2004). These phases include:

- Input Phase

  This phase involves many kinds of data for example names, addresses, demographics and business rules used for data validation. The majority of errors occur in this phase, which also includes data warehouse ETL (Extract Transform Load) processes.

- Transmission Phase

Transmission errors are less common than processing errors. However, data transmitted over public and private networks may be intercepted and tampered with during transmission.

- Processing Phase

Errors occurring in the processing phase result from programming flaws, particularly incorrect logic. While these errors are less frequent than input errors, they can have a greater impact. The effects of processing errors can affect entire files, and if the error is embedded in program logic, the effects may last for long periods of time.

- Stored Data Phase

The major source of stored data errors is during data conversion, when data is transferred from predecessor or legacy systems to new or updated systems.

- Output Phase

Errors in this phase can be the result of the through-flow of earlier errors, as well as information delays and error correction delays. Poor labelling of printed data may cause misunderstanding of a report. Inappropriate aggregation of data can lead to reports containing too much or too little data. Lastly, a lack of user training may result in errors in the output phase.

Integrity impairments may occur in any of the phases mentioned and affect each of the information integrity attributes (completeness, currency/timeliness, accuracy/correctness and validity/authorization). For example, duplicated data is an integrity problem which affects the validity/authorization attribute of data during its input phase. Errors may occur independently in each phase, or they may be cumulative in effect (Boritz, 2004).

As stated previously, according to Boritz (2004), there are four core attributes essential to representational faithfulness:

1. Accuracy/Correctness

Accuracy is about whether information is true to the real-world. For example, if the inventory states that there are three widgets in the warehouse, there should be three widgets in the warehouse.

2. Completeness

Completeness affects the level of accuracy, as this is reduced when the data are not complete. For example, if the database of a telecommunications company only reflects some of the calls made by a party due to processing delays, the total amount of minutes used, and thus, the amount due, would be incorrect.

3. Currency/Timeliness

Information currency is affected by real-world changes over time and processing delays. Currency can affect the accuracy of information, as information which is out of date is possibly not accurate. For example, the data base states that a policy-holder is single, but he has since married after the data were updated.

4. Validity/Authorization

Validity for intangible objects refers to information representing real conditions, rules or relationships, not merely the characteristics of physical objects. Transactions are deemed to be valid if they are initiated and executed by personnel or systems which have the authority to do so. This means that validity is reliant on both accuracy and authorization. For example, the credit limit of a client must be within the company rules and regulations for credit limits, in order to be valid.

These are minimum criteria for assessing the presence of representational faithfulness, and are all necessary within a system, but do not individually assure representational faithfulness. These attributes are affected by a set of seven enablers. The enablers are not characteristics of representational faithfulness, but are essential to achieve it. They are:

1. Security

Physical and logical access controls and safeguards must be implanted to protect information (both in motion and at rest) from acts of nature, intentional malicious acts and errors, all of which could negatively affect integrity.

2. Availability/Accessibility

Information must be complete, current and timely and must be accessible to users, who should be able to retrieve the needed information in a useable form, in accordance with business specifications.

3. Understandability/Granularity/Aggregation

Factors, such as user knowledge, skill, training and motivation, all affect the understandability of information. Design choices, including the level of aggregation (granularity) of the information, also affect the comprehension of information.

4. Consistency/Comparability/Standards

The stability and consistency of a system can be affected by changes caused by environmental uncertainties. If the stability and consistency are affected, the comparability may also be affected. Environmental factors include complexity, change, IT devices and computer crime.

5. Dependency/Predictability

Dependability refers to consistency in how information is measured and presented to decision-makers. Predictability of events can enable representational faithfulness of information.

6. Verifiability/Auditability

Verifiability is the ability of independent observers to obtain the same result when processing information in the same way as that system, in other words to be able

to substantially replicate the results produced by it. Auditability features are those which enable tracing of information to its source and confirmation of representational faithfulness, for example, use of unique transaction/record identifiers.

7. Credibility/assurance.

There needs to be evidence that the information's integrity has not been breached in order for the information to be trustworthy (Boritz, 2004; Boritz, 2005).

5.3.2 Integrity and Security

According to Boritz (2005), information integrity is enhanced by processing integrity, as the level of system integrity limits information integrity. System process integrity depends on system availability and system security, as the information integrity attributes (completeness, currency, accuracy and validity) emanate from the attributes of systems reliability (processing integrity, availability and security). Availability contributes to the completeness, currency/timeliness and accuracy/correctness of processing, and security contributes to completeness, currency/timeliness as well as validity/authorization. Security additionally contributes to availability. Security is, therefore, critical in its role as an enabler of information integrity (Boritz, 2005). These relationships are shown in the following diagram.

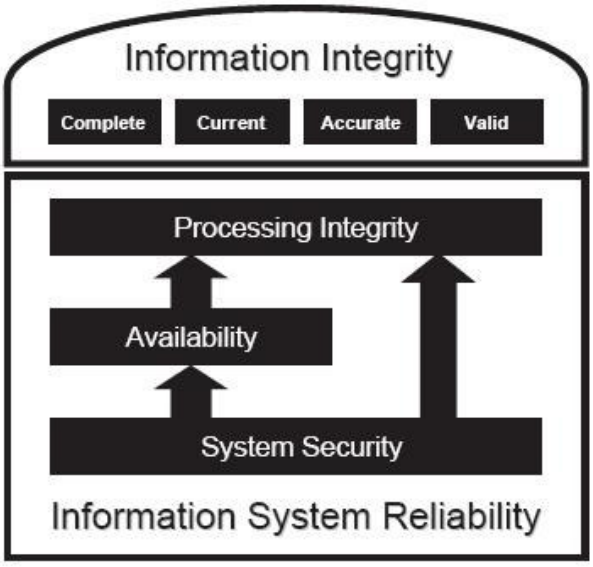


Figure 5.1: Relationship Between Information Integrity, Processing Integrity and Reliability (Boritz, 2005, p. 269)

Many aspects affect information integrity. Each of the technologies in a continuous auditing system must be implemented in consideration of the enablers and in protecting information integrity to increase the trustworthiness of the results produced. The table below shows the elements of continuous auditing systems and some of the suggested measures to protect them.

Elements of Information System	Security Measure
Operating System	Active Directory
	Administrator Password Security
DBMS	Users & profiles
	Roles & privileges
	Administrator Password Security
	Encryption
	Logging and auditing
Log Files	DBMS Utilities (e.g. DataMirror)
	Separate/secure location
	Encryption
Network	VPNs
	Firewalls
Web Services	Encryption
Report (XBRL)	XARL

Securing Alerts (emails etc)	Encryption

Table 5.2: Protecting Technological Elements of a Continuous Auditing System

Operating system

One of the most crucial aspects of security at operating system level is the possession of administrator rights. Anyone with administrator privileges may be able to compromise either the information in the database or applications, and even worse, may be able to remove evidence from log files. On systems such as the AS/400 the Data File Utility (DFU) allows administrators to alter database files with almost no trace, because although the system produces logs, anyone with permission to use the DFU can also tamper with them. The application to which the data belongs will not show any record of this transaction (Onions, 2003). Solutions include the proper separation of duties, having a separate database administrator and a systems administrator.

Database Management Systems (DBMSs)

Problems associated with integrity in the DBMS are closely related to those associated with the operating system. SQL or other database utilities can be used to commit fraud (Mookhey, 2004). This problem is exaggerated by the fact that connecting directly to the database, using SQL, SQL*Plus or plsql, can allow a malicious user to bypass the normal controls within applications which make use of the database. To protect critical information, even from database administrators, encryption may be used. Commercial encryption utilities can be used to protect the confidentiality and integrity of data (Mookhey, 2004).

When installing a database utility, such as Oracle, it is important to follow the security guidelines recommended for each platform (operating system). The proper management of user accounts and profiles is required to ensure security. A dormant account can be used to commit fraud making it difficult, if not impossible, to identify the culprit. Default accounts are often created in installation; these accounts have well-known or weak passwords and should be disabled. To ensure that no dormant accounts exist, tools, such as the `Audit_Last_Logon` script (which is made

available free of charge), can be used to see the date and time of the last logon for each user. Users performing similar functions should be grouped into roles and privileges should be granted to roles instead of user accounts (Mookhey, 2004).

DBMSs such as Oracle, have built-in 'auditing' features. These assist in tracking the actions performed on the database. Basic auditing should occur at all times and, as a bare minimum, include user access, use of system privileges and changes to database structure. This basic form of auditing will not reveal attempted unauthorised access to specific data, but it will give an overview of "incorrect" access and misuse of privileges. This basic auditing may reveal unusual or suspect activities by an employee. Detailed logging of the tables that the suspected employee would access can then be enabled. This is done, because constantly monitoring all changes in all tables is impractical and may severely retard the performance of the database system[8]. However, constant monitoring of data changes in critical tables, such as salaries of an HR database, may be advisable. Besides the built in "audit" commands, features including system triggers and update, delete and insert triggers, fine-grained triggers and system logs can be used for tracing transactions in a database[9] (Finnigan, 2003). It is suggested that the logs should be stored to a location where the database administrator does not have rights, but the OS administrator does, for example, in Oracle (when using built-in auditing functions), the `AUDIT_TRAIL` parameter can be set to `OS` to store the logs in a file within the operating system instead of the database (Mookhey, 2004).

Oracle auditing can be used to create scripts which log database activities and can reveal where further investigation is warranted. Finnigan (2003) presents SQL code to monitor the following activities:

- *Failed logon attempts* – could show attackers' attempts to gain unauthorised access by guessing usernames and passwords.
- *Attempted use of non-existent user account* – if this is repeated from the same terminal, it may show the presence of an attack.

[8] The database system referred to here is *Oracle*
[9] These are features of Oracle; other database management systems may have equivalent functions.

- *Access during unusual hours* – This could be caused by overtime work or maintenance, but may warrant further examination.
- *Users sharing accounts* – simultaneous access to an account from multiple locations may be due to an attack. This may also show multiple users sharing a logon account.
- *Multiple access attempts for different users from the same terminal* – may show a password attack.

The above examples all relate to user activities. Audit trails could also track when objects are created or changed within a database (Finnigan, 2003).

Log Files and Integrity

Logs provide an invaluable source of information during an audit, particularly database logs. It is essential that these logs be secured from tampering by users. Some logs are not always suitable for use as forensic evidence, because they may overwrite themselves over a long period of time to save disk space. Database utilities do exist which create logs which may be admissible as evidence. An example of a commercially available database-logging software utility is DataMirror (Onions, 2003).

Onions (2003) suggests that there are six criteria for maintaining integrity within audit logs. Audit logs:

1. should be computer generated and capture all database changes, including the time, workstation and signed-on user which generated the transaction.
2. need to be immediately encrypted and exported to a secure program.
3. must not overwrite themselves
4. must be tamper-proof, and any attempts to tamper with them need to be logged.
5. need to store all key strokes.
6. must capture all data which is endogenous and exogenous to the system.

Attaining the implementation of these criteria may be hindered by certain factors. The name of the signed-on user may be unreliable because many users know other users' sign-on details. One possible solution is to use bio-metrics to sign a user on. The storage of data related to log files not only increases the resources needed to store logs, but it also means that a flat file would not suffice. Data such as excel spreadsheets, etc., would need to be stored for protracted periods of time. Logs would need to be within a dedicated secure database. ERP software, such as SAP R/3, also support online logging and tracing of transactions, with features such as online auditing and internal-control evaluation tools (Rezaee et al., 2002).

Another form of log suggested is *black-box logging*, so named because it is analogous to the black-box recorder of an aeroplane. This type of log is used for tertiary monitoring of continuous assurance systems. One of the aims of this logging is to limit opportunities for fraud by the collusion of auditors and managers. This enables tertiary monitoring, which is described by Alles, Kogan and Varsarhelyi (2003) as 'the audit of the audit'. Black-box logging makes the manipulations of a continuous assurance system more visible. It is more than merely an extension of the existing practice of documenting audit activities, as it uses a continuous assurance methodology to consistently implement standard control principles, such as, adequate record maintenance, separation of duties and proper authorisation of audit activities. Not only does black-box logging enhance integrity of the continuous auditing system, it forms the foundation for a more thorough and visible system of corporate governance (Alles, Kogan, & Vasarhelyi, 2003)

Networks

Secured transmissions are a requirement within a continuous auditing system. Virtual Private Networks (VPNs) are most often suggested. A VPN functions as a network within a network. VPNs are relatively inexpensive and can transmit data, even across public networks, such as the Internet, safely and securely (Whitman et al., 2003). The use of new paradigms for communication via networks, such as Web services, is becoming more commonplace.

Web Services

Murthy and Groomer (2004) discuss using Web services in conjunction with XBRL to continuously audit business processes. Their model for accomplishing this is known as CAWS (Continuous Auditing Web Services). CAWS can provide assurance about a specific business process by working with a granular level of data. It can also provide assurances on continuously reported earnings, using an aggregate level of data. CAWS is also capable of providing continuous assurance of internal controls within a client's system. All of these assurances can be invoked on command - this is known as a 'pull' model of auditing.

A Web service is basically an application supported by a set of technologies (XML, SOAP, WSDL and UDDI) which supports a business process and can be run over a network. Web services can exchange data with each other and provide data on behalf of programs running on a network. SOAP is the messaging and communications protocol used by Web services to communicate between applications. SOAP messages use XML to represent the transmitted data which is sent via common Internet transport protocols (e.g. HTTP). In order to improve the security of these messages, enhancements to the SOAP protocol have been suggested. "Web Services Security Language" or "WS-Security"[10] was developed to provide integrity, confidentiality and authentication to Web services (Murthy & Groomer, 2004).

Reports (XBRL)

As discussed in Chapter Four, XBRL does not guarantee the quality and reliability of information (Boritz et al., 2003). For example the wrong document taxonomy may be applied to XBRL (a different version), and, although this may affect data integrity, it would be very difficult to detect. Another problem is that XBRL documents are easily created and modified, allowing anyone to 'fake' financial data. To overcome these problems and help to secure XBRL reports, XARL can be used, as it enhances XBRL by providing assurance that the data is credible. It would, for instance, prove that the document was audited or assured by a public accountant (Alles et al., 2004; Boritz et

[10]More information available at http://www-106.ibm.com/developerworks/webservices/library/ws-secure/ (last accessed may 2005)

al., 2003). XARL does this through a combination of public-key cryptography and XML encryption.

Securing Alerts

It may also be desirable to prove the authenticity of alerts and messages sent out by the system. This can be done through public key architecture.

5.4 Conclusion

The concepts of integrity and quality were emphasized in this chapter. It was established that data quality refers to data which adequately suits the needs of data consumers. A high degree of data quality is important, because poor data quality drastically affects the effectiveness of an organisation and can cause time-delays which are unacceptable in a real-time system. Furthermore, even the appearance of poor data quality can hamper the decision-making process, which continuous auditing aims to support (Redman, 1998). While integrity can be simply defined as the state of being error free and uncorrupted, this chapter created a more comprehensive definition including the concept of representational faithfulness (Boritz, 2004). Data integrity and systems integrity were also differentiated.

This chapter aimed to explain quality and integrity, in order to clarify the aims of continuous auditing, particularly the ultimate aim of continuous auditing systems, which is to use technology to detect and prevent fraudulent and erroneous transactions from being processed. It is essential to verify the integrity of data before any of the audit objectives can be fulfilled, as it is essential that the conclusions in auditors' reports are based on accurate and reliable data (Wessmiller, 2002).

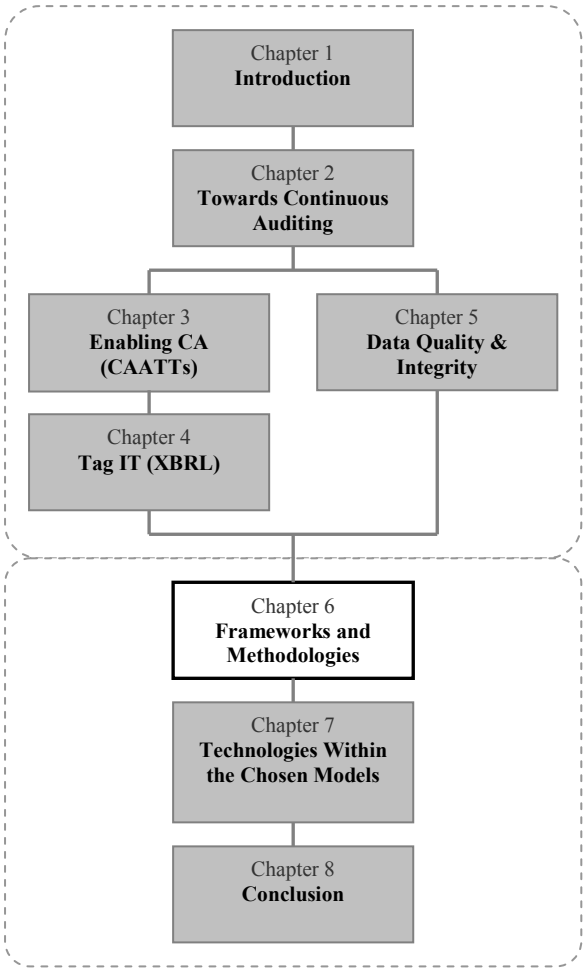
Information and data security were shown to be of critical importance in the role of an enabler of information integrity (Boritz, 2005). Security measures are required to be embedded in elements of a continuous auditing system, specific examples of security measures were discussed.

Now that the topics of integrity, quality and security have been introduced, the following chapters will deal with continuous auditing models. The next chapter will introduce the three models of continuous auditing.

PART II – Solution Chapters

Chapter 6

Continuous Auditing Models



Chapter 6:

6.1 Introduction

6.2 Three Models

6.2.1 Rezaee et al. A Continuous Auditing Approach (2002)

6.2.2 Onion's Proposed Model for Secure Continuous Auditing (2003)

6.2.3 Woodroof and Searcy Continuous Audit: Model Development and Implementation Within a Debt Covenant Compliance Domain (2001)

6.3 Conclusion

6.1 Introduction

In this chapter, three continuous auditing models will be explored. These will be discussed in relation to their aims/foci, component parts and the steps involved in implementing each model. Different approaches and methods for performing continuous auditing exist. This is due to the level of automation, introduced as a result of the system design and implementation. The majority of approaches rely on highly automated processes. These processes use embedded audit modules to constantly monitor and report on significant audit events within data. Less automated processes involve the capture and transformation of transaction data. Some auditor intervention

may still be required to query unusual patterns and isolate exceptions (Rezaee et al., 2002).

Most suggested continuous auditing models are merely conceptual. Few seem to have been implemented in real-time systems. Possibly the first continuous auditing system to be published was in 1989: *Continuous Auditing of Database Applications: An Embedded Audit Module Approach (Groomer et al., 1989).* One of the early models to be implemented was the *Continuous Process Auditing System* (CPAS) which was developed at AT&T Bell Laboratories. This is a methodology for internal auditing of large 'paperless' real-time systems (Vasarhelyi & Halper, 1991). This model appears to have formed a basis for the later models.

6.2 Three Models

Three of the better known models have been chosen for discussion. The differing approaches and technologies will be discussed. These recent models include "a continuous auditing approach" discussed in an article entitled "Continuous Auditing: Building Auditing Capability" (Rezaee et al., 2002), "A Model for Secure Continuous Auditing" from the article "Towards a Paradigm for Continuous Auditing" (Onions, 2003) and "Continuous Audit: Model Development and Implementation within a Debt Covenant Compliance Domain (Woodroof & Searcy, 2001).

6.2.1 Rezaee et al. A Continuous Auditing Approach (2002)

A conceptual framework, described as a continuous auditing approach, is laid out within a paper which describes continuous auditing methodologies. The definition of continuous auditing given for that paper is, "a comprehensive electronic audit process that enables auditors to provide some degree of assurance on continuous information simultaneously with, or shortly after, the disclosure of the information"

Rezaee states that one of the most complex and challenging aspects of building continuous auditing capability is the standardisation of data. Diverse file types and various record formats produced by various sources, including legacy systems, must be catered for. This creates a risk of duplicating records and introducing errors, which could negatively affect the integrity of the information.

Within this chapter, two approaches are described for the collection and storage of data. The use of a 'scalable audit data warehouse' is the first approach. A repository for storing transaction data produced by different business systems (data warehouse) is designed. This data warehouse should be scalable; allowing for increasing amounts of data as audits progress. This approach is recommended for dispersed systems with varied data formats. This approach can be expensive, particularly when embedded audit modules are required within complex legacy systems. In such cases, the second approach may be more beneficial. Instead of building a data warehouse, subject-specific audit data marts are used to automate the capture of relevant data, and the auditing and reporting processes.

The continuous auditing solution could be capable of running on a distributed client/server network and is also Web-enabled for transmitting data to audit workstations. The model is executed as follows:

1. The data is collected from transactional systems. This is done by linking to tables, via FTP, storage drives or modems. The data is then stored on an audit server.

2. Once on the audit server, data is extracted from a variety of platforms and systems. Data standardisation is therefore required. Standards and formats are developed for storing data in the data warehouse/mart. The data is then transformed by cleaning, validating, restructuring the data and 'scrubbing' with business rules.

3. An enterprise-wide data warehouse is not always needed, as it may be too expensive and complex. Instead, the required data could automatically be fed into several data marts. The data marts contain metadata, which details the source transactions and the ETL (Extract Transform Load) process, as well as the tests which take place. The metadata may, for example, include detailed file definitions, business rules and transaction-process flows.

4. Standardised tests are created to run within a data mart. The tests are created either to run continuously or at predetermined intervals. The tests are designed to automatically gather evidence and issue exception reports.

The model illustrated below does not require an enterprise-wide data warehouse. In this solution, many subject-specific data marts are automatically fed data, and periodically selected data is extracted. This data is selected according to an audit-testing plan. This reduces the number of elements needing data transformation and mapping

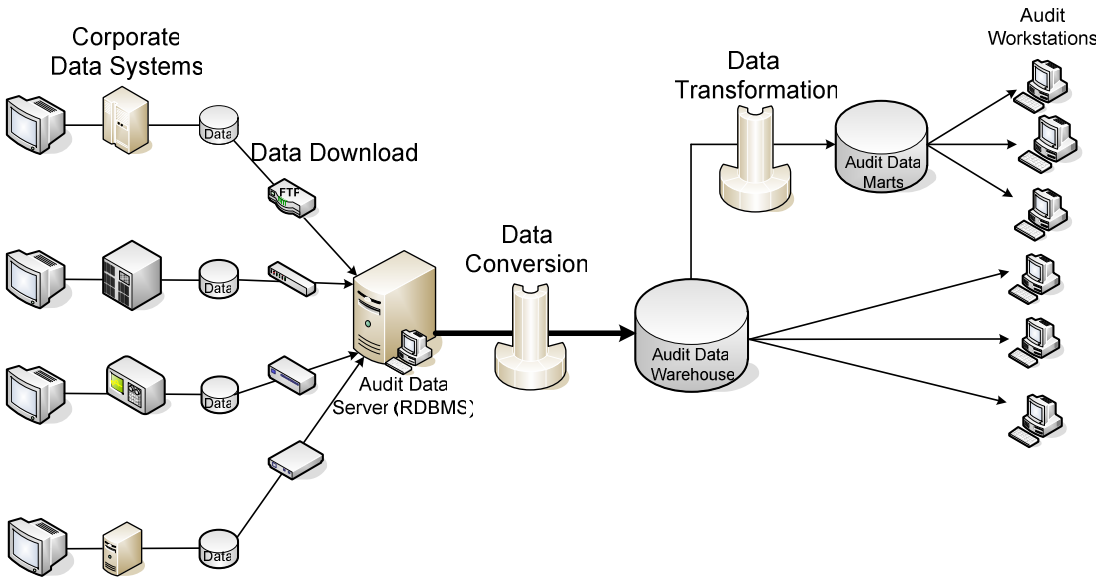


Figure 6.1: Continuous Auditing Approach (Rezaee et al., 2002, p. 156)

The model consists of the following components:

- ***Corporate Data Systems***

  Includes enterprise computing solutions such as SAP R/3, BaaN, PeopleSoft, Oracle or SQL. The data may be in various formats, such as VSAM, IMS, ASCII, MDB, CSV, XLS or TXT.

- ***Audit Data Server***

  To aid easier access, analysis and reporting, the data collected for various business units' data marts are physically stored in the audit data server.

- ***Audit Data Warehouse***

  Once converted, selected transactions which are deemed to pose an audit risk are collected and stored in the audit data warehouse.

- ***Audit Data Marts***

  Standard metadata, containing the complete details of source transactions and the Extract Transform Load process (e.g., file definitions, business rules, transaction process flows) are stored in the relevant data mart. A data mart may be for a particular business unit or several highly interrelated business units.

- ***Audit Workstations***

  End users interact with the data mart via audit workstations. Users include auditors, business\unit managers and corporate security officers. There are two generic categories of end-users described. Oversight users only need to access exception reports. Analytical users formulate their own queries and need to interact with the data. This requires advanced data extraction and analysis software on the audit workstation.

This model has the widest scope of the three models, and it is depicted as a relatively generic solution to continuous auditing. It also seems to focus on the standardisation of data. The next model focuses more on information integrity and secure continuous auditing.

6.2.2 Onion's Proposed Model for Secure Continuous Auditing (2003)

The aim of this model is to introduce the concept of continuous auditing being a new paradigm. It suggests that in order to guarantee the integrity of accounting information captured in ledgers, it is necessary to monitor all keystrokes and transactions within the system, and then search for patterns in groups of transactions. Expert systems are suggested as the means by which to search transactions for patterns. These actions concern the three basic areas of data examination within a continuous auditing system. These are keystroke level, transaction level and transaction pattern level data examinations (Srinivas, 2006b). All three areas of data

examination are required to make the continuous auditing system both effective and comprehensive.

The first of these areas, the keystroke level data examination, basically involves monitoring database utilities and applications for commands which could cause fraud or error. Database utilities can allow users to modify and update a master file, bypassing the normal safeguards present in the accounting system itself. Committing this type of fraud normally entails using a Database Management System (DBMS) and Structured Query Language (SQL) commands or other database interfaces, such as the Data File Utility (DFU) on the IBM AS/400. A fraudster would need to logon using a supervisor/administrator password, which is possible if security of these database utilities is overlooked. In many organisations, administrator passwords are known to many people, or are not regularly changed, or are never even changed from the defaults created during installation.

Transaction level data examination involves auditing and reporting on each transaction as an isolated entity. This is done 'ephemerally'- the transactions are tested at the time of entry. It ascertains whether each transaction meets the pre-specified rules for transactions. These may be business rules, or even rules dictating what actions are permissible for certain users. This is done in conjunction with performing certain analytical functions. Computer Assisted Audit Tools (CAATs) could be used. These operations would need to be performed on the transactions in real-time, rather than batch-processing. Most of the current CAATs run in batch and would have to be modified.

After the transaction has been examined, it may be added to a data mine for possible further examination. Transaction pattern level examination involves examining the effects of transactions, as a whole, over a longer time period (perhaps even years). Expert systems and rules-based criteria are used to identify patterns in the groups of transactions which could together result in fraud. The rules would be similar to virus definitions and would be made available for different industry types.

The problem when attempting to use expert systems is that each available software package has a different data schema. It would be very costly and time-consuming to

create expert systems for each application. The suggested solution is to create a generic master file and transaction layout which could be used, regardless of application data schema. This newly defined generic schema for a transaction would allow one expert system to trawl through the data mine. This schema could be defined using eXtensible Continuous Auditing Language (XCAL). XCAL which like XBRL, is XML-based. One advantage of XCAL is affordability, making it useful for SMMEs.

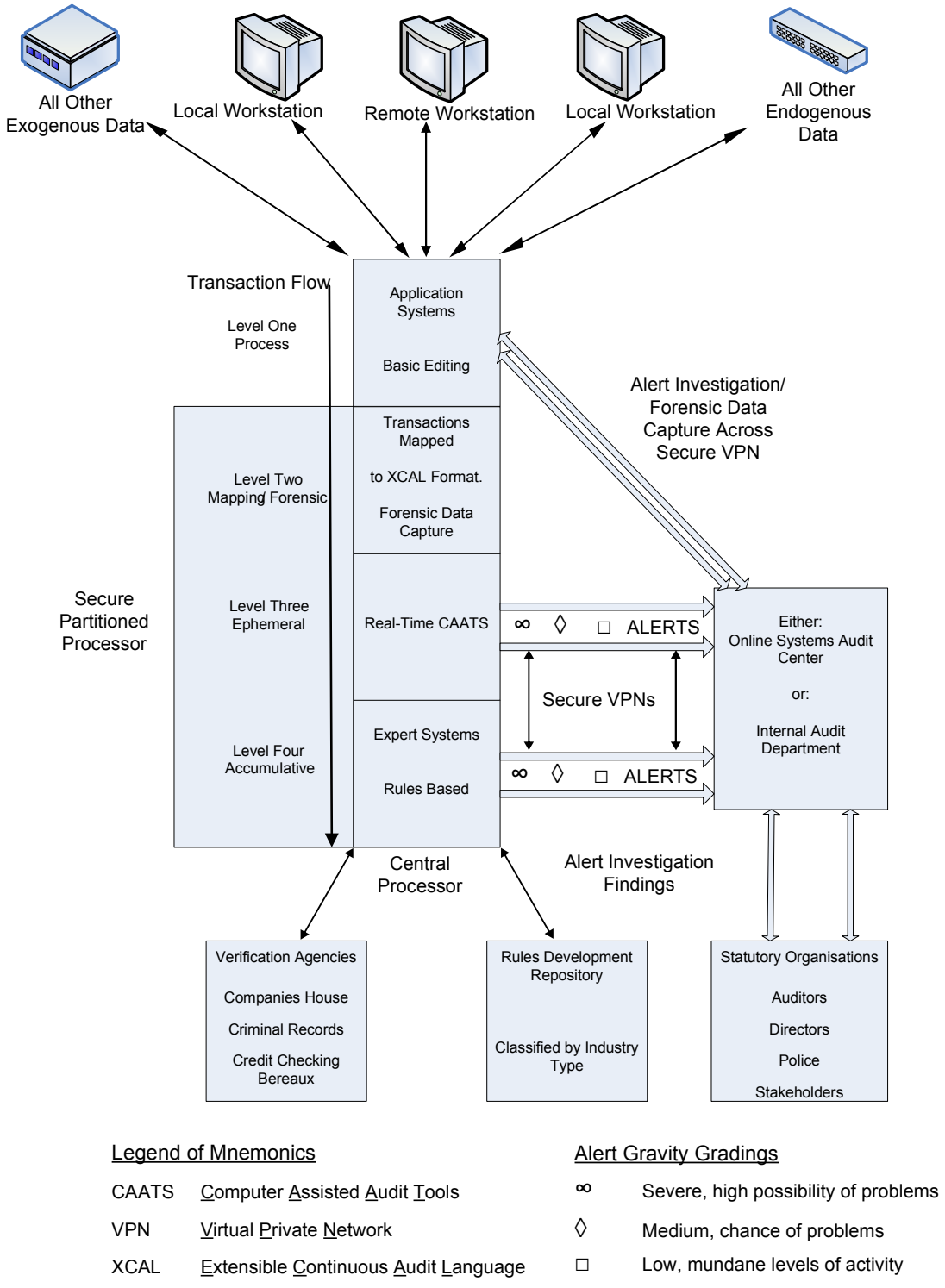


Figure 6.2: Onion's Model for Continuous Auditing (Onions, 2003)

As shown in the diagram, the model consists of four ‘levels’ which describe the steps involved. These levels describe the steps taken when the model is running and are not the same as the data levels mentioned previously. The previously mentioned levels are the three areas of examination for the model to be comprehensive and effective.

The steps involved in this model (referred to as levels) are:

1. Transactions and data from various sources are entered for processing.
   Basic editing takes place within the application. The applications appear to run as normal, without delay.

   The three remaining levels are partitioned and secured from all users (including administrators). These are installed and maintained by government approved Systems Audit Centres.

2. Transactions and keystrokes are mapped to XCAL schemas. This is done in real-time. Transactions are captured forensically on a daily basis, so that they may be submitted as evidence should fraud be detected. DBMS utilities are used to capture the data to a secured storage medium. In the UK, data needs to be exactly as the user entered it, in order to be admissible as evidence, thus the data may not be encrypted when captured forensically.

3. Real-time CAATT processing is used to check transactions and keystrokes. This runs slightly after the application (first) level, but only by nanoseconds. Rules for this stage of auditing will vary according to transaction types. They are compiled from knowledge gathered from experienced auditors and forensic accountants. These rules may enforce such issues as acceptable business days and hours and acceptable batch costs.

   Alerts may be sent to an Online Systems Audit Centre (OLSAC) via secures VPNs (Virtual Private Networks). OLSAC is a group of professionals, skilled in information systems, business processes and auditing which monitor alerts and investigate them online. OLSAC is authorised to do this by government or professional accounting bodies. OLSAC methodologies and techniques differ

from current audit methodologies, for example, the current auditors would still prepare the annual financial audit report, to which an OLSAC certificate is appended. This certificate confirms that the systems have been operated and investigated according to current standards. It would also detail the alerts of the past year, and the actions taken as a result of these alerts.

The alerts, which are sent out at level three, are graded according to levels of gravity. Transactions are stored at this level for a day, after which they pass to level four, where they are stored for years.

4. Expert systems look for patterns in the data, which are pre-defined by expert rules. These rules are formulated from knowledge elicited from experts' experience, standards, laws, best practices and historic transaction patterns. Artificial intelligence within an expert system will also allow new heuristic rule sets to develop. These are automatically included into the ongoing analysis of the expert system. All the rules are aggregated together in a single separate knowledge entity, known as the 'rules repository'. Here, the expert system stores as much information as possible about a given subject, in this case business rules, which are used in conjunction with heuristics to analyse transactions. Newly formulated rules are delivered and installed, via the Internet, with release level techniques, much like antivirus updates. Rules are classified by industry type, as industries may have subtle differences, although the financial ledger systems should be similar.

   The knowledge gained from experts is continually analysed by asking thousands of "`if then`" questions. These questions relate to both single transactions and groups of transactions.

   The alert processing system of level four is similar to that of level three. Alerts are also graded, but level-four alerts are more likely to be complex in nature. The alerts are sent to auditors over secure networks, such as VPNs.

   In level four, the system would also have the ability to communicate with various agencies in order to verify data from transactions. These verification

agencies may include criminal record databases, Companies House and various credit bureaux. An example where fraud could be detected in this way is where new supplier details are checked against Companies House. Post codes or directors' names corresponding to those of employees may reveal fraud. This ability to communicate with external agencies is facilitated by Web Services. Web services use XML to exchange data and parameters between software applications, via a network, such as the Internet or a VPN.

This model focuses primarily on data integrity. It also attempts to find a solution to solving the problem created by trying to use expert systems where a variety of data formats is present. In the next section, the last of the three models is discussed.

6.2.3 Woodroof and Searcy Continuous Audit: Model Development and Implementation Within a Debt Covenant Compliance Domain (2001)

The focus of this model is to create a working model of continuous auditing. This was done for one particular business domain. The definition of continuous auditing used in this paper is, *"an assurance service, where the time between the occurrence of events underlying the particular subject matter of a client, and the issuance of an auditor's opinion on the fairness of the client's representation of the subject matter is eliminated"*.

A conceptual model of continuous auditing is proposed in this paper. This model is limited; it is discussed in relation to debt covenant compliance. The model makes use of Web-enabled technologies. It draws attention to the need for a reliable and secure system. The need for the production of 'evergreen reports' is also discussed. Evergreen reports are reports which are generated on demand, usually viewed through a Website. The model is based on a database of transactions (journals and ledgers) on the client's system, with a Web interface (on the auditor's system) for the auditor to use.

A debt covenant is the legal agreement between a lender and a borrower. It specifies the terms of the loan (the duration, rate and payment information), as well as required collateral and what values of key variables constitute compliance (and how to remedy

non-compliance). The model allows compliance with the debt-covenant domain to be monitored via the Web by the lender. Loan officers are presented with a Webpage listing all loans for which they are responsible, and the debt-covenant agreement and the relevant criteria related to the loan. This allows a loan officer to continuously monitor whether the actual values of the client's variables are in compliance with those in the covenant agreement.

The steps involved in this model are known as 'stages'. The model is implemented in five stages:

1. The loan officer requests an evergreen report by sending the client's loan covenant parameters to the auditor.

2. Agents and sensors within the client's system monitor the transaction data for exceptions to pre-specified rules. These exceptions may trigger alarms, and alerts are sent to the auditor. The rules check the reliability of the system (possibly using continuous SYSTRUST), the fairness of the representation of financial reports and compliance to third-party contracts (like debt-covenant agreements).

3. A *digital agent* on the auditor's system requests a digital agent on the client's system to retrieve the client's real-time balances of accounts, via stored procedures in the client database.

   In the context of this model, a digital agent is a set of electronic instructions (software) which acts on behalf of the auditor in a semi-autonomous manner to perform some service related to the subject mater being audited. Using digital agents allow auditors to use information which is published on the Web.

   The client's digital agent creates a backend page from the information retrieved by the stored procedure. These pages are never called directly and do not have a user interface, they exist only to facilitate the use of digital agents.

4. If more information is returned than is needed in the backend pages, the digital agent extracts the information relevant to the contract (in this case the debt-covenant compliance). This is done in a temporary workspace which is parsed by the digital agent for values relevant to the debt covenant compliance. The information is checked for compliance, the actual event or process is checked against an acceptable standard for that event or process. If anomalies occur, these are flagged, and the auditor is notified so that he/she may take action.

5. An evergreen report is generated and displayed to the loan officer. This details three levels of assurance. Level 1 is an assurance of reliability. If there is level 1 exception, no further analysis is performed. Level 2 offers an opinion on the fairness of real-time financial statements. Level 3 provides an analysis of technical violations of third-party contracts (in this case, debt-covenant compliance is assessed).

Due to the reports being pulled (produced on demand), as opposed to being pushed to the user, this model is less suited to using XBRL-based reporting.

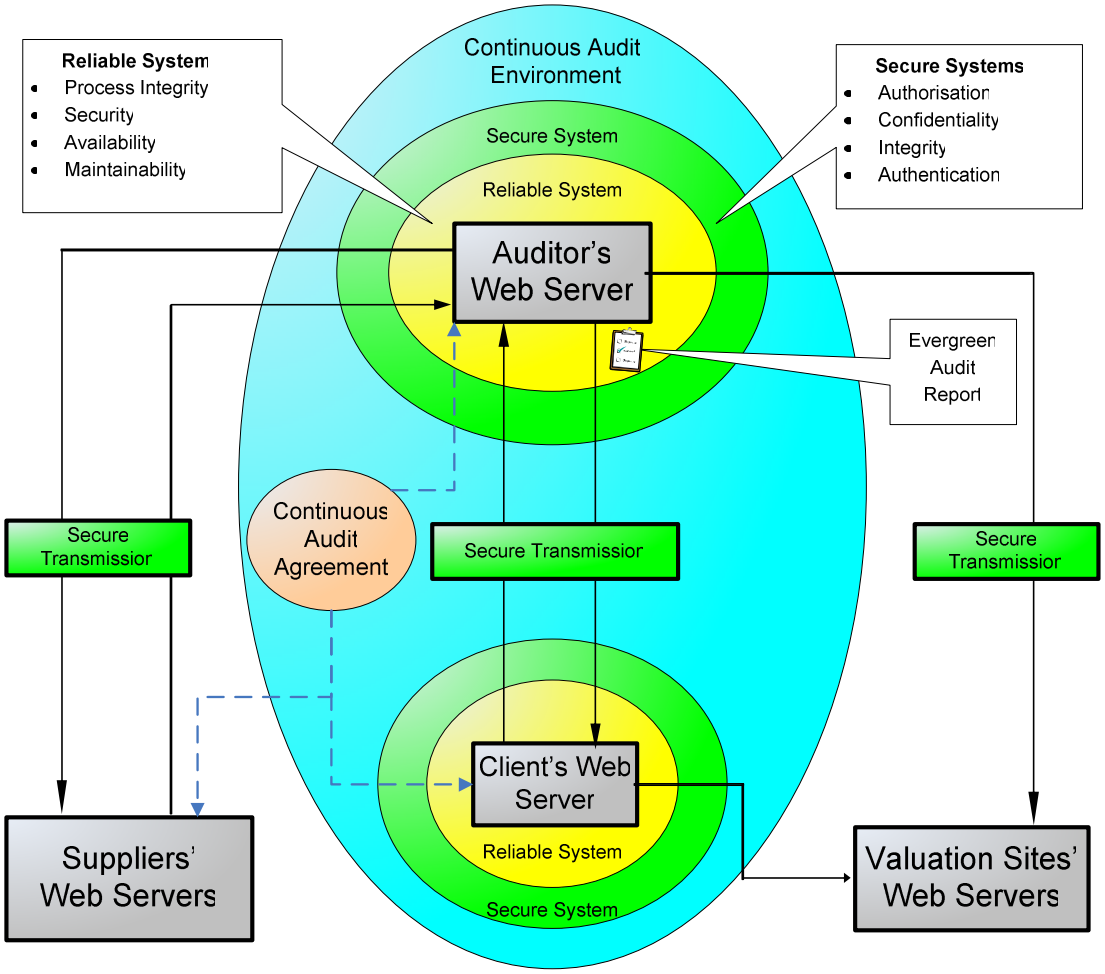


Figure 6.3: The Conceptual Model of a Continuous Audit (Woodroof & Searcy, 2001, p. 22)

The components of continuous auditing, as depicted above, include:

- ***Various interconnected Web servers:***

  The participating Web servers are interconnected and are given authority to communicate. The auditor has controlled access to the client's database through the client's Web server. Approved third parties and external users have limited and controlled access to the client's database through the auditor's Web server, in turn. Third parties are suppliers and customers, who have agreed to a continuous auditing relationship with the client through the auditor. These third parties may represent customers, vendors (suppliers of merchandise) and financial institutions (suppliers of capital). Automatic electronic confirmation of account balances, such as accounts payable, cash and accounts receivable would be possible. The valuation site's Web server

represents Websites containing information on adjustments made to various accounts, for example, the market prices for stocks held for resale.

- ***Continuous audit environment:***
  This is described as, the data flowing through the client's system is continuously monitored and analysed using devices integrated within the system.

  Auditors are notified via the Internet of exceptions to auditor-defined rules when alarms are triggered. The alarms notify the auditor of potential deterioration or anomalies in the client's system.

  The aim is to collect information (for example, market values and estimates) in real time so that real-time assurances can be provided by reflecting real-time information in the reports produced.

  Three levels of assurance are provided, each having a different degree of significance and requiring the auditor to take different actions. Level 1 is assurance regarding the reliability of the client's system and the security of transmitted data. Level 2 is an opinion on the fairness of the real-time financial statements produced by continuous assurance. Level 3 concerns the client's compliance to a domain-specific agreement with a third party, as detailed in the continuous-audit agreement.

- ***Continuous assurance agreement between all involved parties:***
  This agreement is a contract between all parties involved, including the audit firm, client, suppliers and customers. The primary parties in this contract are the audit firm and client. The contract should also outline the responsibilities of the client's suppliers and customers in executing continuous-audit routines. The agreement should also detail technicalities surrounding the continuous audit, such as system accessibility and availability to the auditor and the handling of exceptions flagged during the audit.

- ***Characteristics of a reliable system:***

  Continuous assurance is considered worthless if the underlying reliability of the systems producing the assurance is in doubt. An adaptation of SYSTRUST from a periodic assurance to a continuous assurance is suggested.

  According to SYSTRUST reliability covers the four principles of integrity, security, availability and maintainability.

- ***Characteristics of a secure system:***

  All transmission of information between parties should be authorised and be confidential, have integrity and be authenticated.

  There may be other reliability and security concerns, which fall outside the continuous auditing environment, yet impact on it. These include the reliability and security of Web valuation sites referenced in the audit, and reliability and security of the (Internet Service Providers) ISPs. These concerns may be addressed by services like Better Business Bureau, TRUST-e, Veri-Sign, international computer security associations and WebTrust.

- ***Evergreen reports:***

  These are the dynamically dated audit reports available to the user through a Webpage within the continuous-auditing environment.

These components together form the basis of this model. This model differs from the first two in that it functions in a limited business domain (debt-covenant compliance) and it is Web-centric. This model, however, could be altered to suit other third party agreements, and it would possibly be applicable to other business domains.

A summary is tabulated in the following table, which shows the technologies found within each model. These have been grouped according to the similarity of their purpose in each respective model.

Rezaee	Onions	Woodroof & Searcy
FTP, storage drives or data transfer via modem	VPN	Web services (XML data) Digital agents
Corporate Data Systems Audit Data Server Audit Data Warehouse Audit Data Marts Audit Workstations	DBMS XCAL Data Mine	Client's Web server Auditor's Web server Valuation site's Web servers
Standardised tests (CAATTs)	CAATTs Expert Systems	Integrated monitoring (possibly EAMs)

Table 6.1: Some Technological Components of the Three Models

6.3 Conclusion

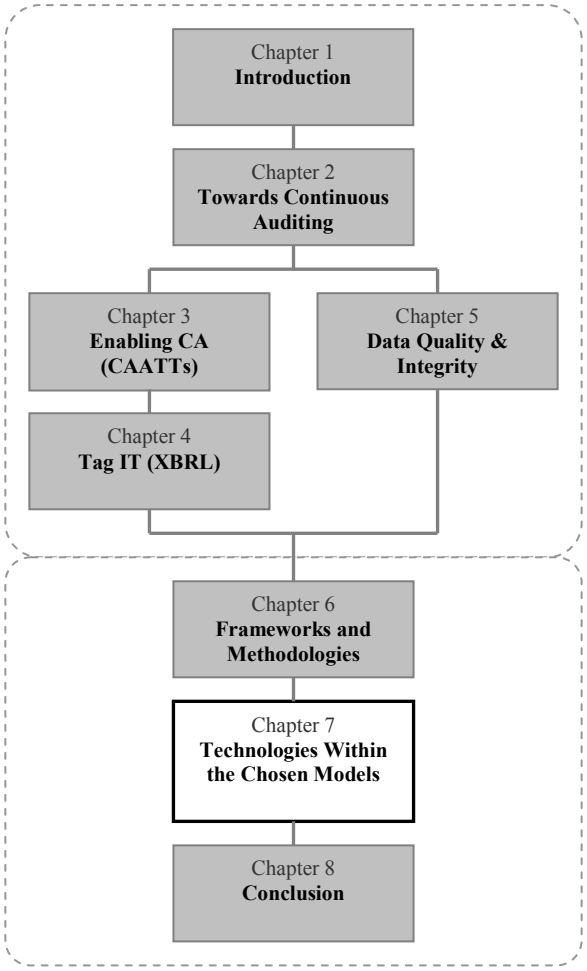
This chapter discussed three of the most influential continuous auditing models. Each has a specific focus or aim. It appears that the more generic model by Rezaee et al aimed to firstly discuss continuous audit methodology and secondly to address the problem of data standardisation. While Onions focuses more on preserving the integrity of data, and overcoming the problem of data formats, for which XCAL is suggested. The focus of Woodroof and Searcy's model seems to be developing a working prototype of a continuous-auditing system, and so the scope of the model was limited to a single business situation, that of debt-covenant compliance.

In this chapter, the functioning of each model was explained in terms of the steps involved. The component parts of each model were also listed as they were found in the source literature. However, these components were not all the technological components used within the models.

The next chapter will build on the findings of this chapter, comparisons will be drawn between these models, and commonalities between the technologies within the models will be discussed.

Chapter 7

Technologies Within the Chosen Models



Chapter 7:

7.1 Introduction

7.2 Evaluation and Comparison of Models

- **7.2.1 Accuracy**
- **7.2.2 Reliability**
- **7.2.3 Real-Time**
- **7.2.4 Reporting Method**
- **7.2.5 Data Acquisition and Solving the Data Format Problem**

7.3 Conclusion

7.1 Introduction

In the previous chapter three models of continuous auditing were explored, this chapter builds on an understanding of the three models and evaluates and compares these models. To evaluate these models, one needs to look at how accuracy and reliability are validated. In this context, *Accuracy* refers to how fraud and error in transactions are detected and how possible material misstatements in financial records are detected. *Reliability* is how confidentiality, integrity and availability of internal controls are examined. Thus, in this chapter the three models are examined with regards to how they validate accuracy and reliability. Other important aspects of continuous auditing models are also compared. These include the real-time nature of each model, the reporting method used and how the problem of differing data formats

is addressed. This comparison is presented in tabulated form. It was published as part of an article and is included as Appendix A. After the table is presented, each of the comparison's criterions is elaborated.

7.2 Evaluation and Comparison of Models

	(Rezaee et al., 2002)	**(Onions, 2003)**	**(Woodroof et al., 2001)**
Accuracy **(Fraud and Error) within Transactions**	Standardised audit tests are built into audit data marts. They run either continuously or at predetermined times. These gather evidence and then generate the relevant reports.	Transactions are checked both at time of entry and later. CAATTs (Real-time, not batch) Expert systems (not in real-time but running continually)	Rule-based detection by digital agents. Data is analysed by devices integrated into the system.
Reliability **System of Internal Controls**	CAATTs are used. These include Integrated Test Facilities and Parallel Simulation. ITFs are used to verify correctness and completeness of processing. Parallel Simulation tests assess effectiveness of control activities.	Parsing of keystrokes to detect database management utilities. Password control. Operating system's security. Audit logs. Web services verify information (e.g. new supplier's credit history checked).	Adapt and apply SYSTRUST principles. Web-based valuation sites. Must be in the auditor defined rules for the Digital Agents.
Real-Time	Real-time processing is the aim for this system.	All proposed systems run in parallel with operational systems in real-time.	Real-time reporting is one of the aims of this model. To this end, information must be collected and monitored in real-time.
Reporting Method	Web-enabled data delivery of data to auditors' workstations, where reports can be generated (possibly by GAS).	Graded alerts sent through VPNs to audit department/OLSAC. The alerts are graded by gravity (3 levels).	Three levels of reporting, and alerts are sent to the auditor via email. Level 1: reliability of the system or security of the transmission. Level 2: transactions and processes. Level 3: technical violation of 3rd party agreement. 3rd party and auditor notified by email. Evergreen reports are produced on demand through a web interface - Information *pull* approach. (As opposed to XBRL reporting this is push reporting method).

Solution to Data Format Problem	Data Mart Data Warehouse XBRL	XCAL Data Marts	Does not interface with legacy systems

Table 7.1: Comparison of Three Continuous Auditing Models (Flowerday et al., 2006)

In order to meet the requirements of continuous auditing, any comprehensive continuous auditing model would need to address both *internal control testing* and *testing of transactions*. A dual-pronged approach, where both the system (internal controls) and data (transactions) are tested simultaneously, is required.

7.2.1 Accuracy

Once data have been collected and transformed, the transactions need to be assessed for the existence of fraud and error. Transactions are individually assessed for integrity (errors and fraud) and validity (business rules).

Possibly the most common way for validating accuracy is by using CAATTs. CAATTs can be made to run as close to real-time as possible, and functions while the data is flowing through the application system (Onions, 2003). Both analytical procedures and substantive testing should be applied to look for fraud and error. Common IT-based fraud schemes often involve the billing system, payroll system and cheque tampering. These schemes often need to be identified at the transaction-data level, as they can depend on groups of transactions within the system. Examples include ghost vendors, ghost employees and exploiting voids and returns (Taylor, 2005).

Database management systems (DBMSs), for example, Oracle, can also be considered a type of CAATT (Champlain, 2003). Oracle allows ‘triggers’ to perform tasks when certain criteria are met. Triggers are useful for creating logs of system events (Finnigan, 2003). Triggers in a database can be used in the same way as data query modules (DQMs). DQMs are macros or programs which perform queries to answer a specific question, built using audit software. An example, where DQMs could be used, is to look for fraudulent travel-allowance claims by employees. Employee swipe-card data could be compared to dates on tour and travel reports. If the employee’s card was swiped, and they were at the office, they are most likely

claiming travel allowance for extra days (Dalal, 2000). If the entry of a new travel claim triggered the execution of the relevant DQM, an instant audit would occur.

Once the transaction data has been examined by CAATTs (including DQMs) and is captured within the DBMS, alerts could be produced. For example, control agents may be used to alert auditors, if transaction values change too much from the norm. The relevant transaction data then needs to be collected for future forensic analysis - possibly by moving the data to a secured partition or dedicated audit server. This may be achieved using FTP (File Transfer Protocol) and tape or other large-capacity storage devices.

Digital agents then examine transactions and select those that should be set aside for later analysis. CIS (Continuous and Intermittent Simulation) could also be used for deciding which transactions require more examination. Once the data has been standardised and stored, it may then be checked for groups of transactions and the cumulative effects of a series of transactions. Expert systems and digital agents may be very useful for this purpose. Analytical procedures could also be used to create 'norms', which can be used as a benchmark.

Below is a summary of some of the technologies used in verifying accuracy (detecting fraud and error within transactions). This shows the use of each technology within the three models being discussed.

Technology:	Use in Continuous Auditing System:
Standardised tests in the data mart	Gathers evidence and generates reports (Woodroof & Searcy, 2001)
CAATTs	CAATTs used to perform test of transactions throughout the year, reducing expensive substantive tests of account balances done after balance sheet date. (Rezaee 2002) Auditor designed tools or commercial CAATTs (GAS) Extract data, download information for analytical review, footing ledgers, counting records, identify unusual transactions (Rezaee, 2002, p.151) EAMS: monitor & report events of significance Used for, amongst other purposes, transaction level data examination – examining files and fields in detail, creating tests to verify data, for example, searching for gaps, running Benford's Law on order numbers, creating statistics, verifying that there are the correct number of records, examining data ranges, stratifying data, ageing data, classifying data, creating trends and

	averages.
Expert Systems	Analyse patterns & latency of transactions (Onions, 2003)
Digital Agents	Leverage information published on the Internet. Communicates with client's database, search networks. Capable of determining appropriate audit routines and acceptable errors, run audit routines and generate reports. (Woodroof & Searcy, 2001, p. 11)

Table 7.2: Technologies Used in Verifying Accuracy

As can be seen in the table above, CAATTs are versatile tools, which perform a wide variety of tasks in the continuous auditing environment. GAS tools and embedded audit modules are well suited for examining transactions. Onions argues that CAATTs can be made to run in real-time. Using CAATTs at the auditors' command would provide results possibly too late to avoid the repercussions of fraud or error. CAATTs also play a role in verifying the reliability of the system. This is explained in the following section.

7.2.2 Reliability

There is a need to collect evidence on the quality and integrity of an electronic system in producing reliable and accurate financial information. Technology must aid in verifying the integrity of data, because the conclusions in auditors' reports must be based on accurate and reliable data in order to be deemed trustworthy (Wessmiller, 2002).

There is also a need to ensure security of the system. A system, which is not secure, is not reliable. Ensuring security would involve examining internal controls. If the system is not reliable, the results from that system may not be viewed as trustworthy. Woodroof and Searcy (2001) suggest SYSTRUST (or a continuous auditing derivative of SYSTRUST). COBIT Guidelines, in conjunction with ISO 17799, could also be used. COBIT could address internal control-related issues. ISO 17799 would aid in addressing information security issues (ISO/IEC 17799, 2000; The IT Governance Institute, 2005).

When enforcing reliability through a system of internal controls, the aim is to provide confidentiality, integrity and availability to the data within the system. CAATTs such as parallel simulation and integrated test facilities (ITFs), can be used to achieve this.

In the model by Rezaee et al (2002), concurrent audit techniques for testing effectiveness of a client's internal controls are mentioned. Concurrent audit techniques include SCARF (Systems Control and Review Facility) and the snapshot approach. SCARF functions as an exceptions reporting system. It captures transactions meeting certain criteria (defined by the auditor) by using embedded audit modules. The captured transactions are set aside for later review by an auditor. Embedded audit modules and the SCARF approach could be used to create alerts regarding the status of internal control systems by checking if controls are implemented (Rezaee et al., 2002).

Summarised below are the technologies used to examine the confidentiality, integrity and availability of internal controls, in order to verify their reliability.

Technology:	Use in Continuous Auditing System:
CAATTs:	
	Can be used when selecting samples for tests of controls to identify exceptions and perform confirmations (Rezaee, 2002, p. 151).
ITF	Can be used to determine whether the RTA system is correctly processing valid and invalid transactions and *verifying correctness and completeness of processing.*
Parallel Simulation	Replicating some part of a client's application system to assess the *effectiveness of control activities*
Concurrent processing audit modules	Incorporated directly into important computer applications to continuously select & monitor the processing of data
CIS	Used to select transactions during processing for audit review & provide an online auditing capability. (Rezaee, 2002, p. 154)
Parsing keystrokes	Checking if operating system utilities (e.g. IBM DFU)/DBMSs have been used to add, change, delete data. (Onions, 2003)
OS Security & Passwords	
Audit logs	Forms forensic evidence. Can help when tracing transactions.
Web Services	Verify information with outside agencies
SYSTRUST, WEBTRUST, eSAC, COSO, COBIT	Provide standards/objectives/checklists for internal controls
Digital Agents	Leverage information published on the Internet. Communicates with client's database, search networks Determine control status and generate reports. (Woodroof & Searcy,2001, p. 11)

Table 7.3: Technologies Used to Verify the Reliability of Internal Controls

The table above shows the uses of the various CAATTs used most frequently in verifying the reliability of the internal controls within an audited system. These include ITFs, parallel simulation, concurrent processing audit modules and CIS. Tests to verify the reliability of internal controls should be performed on an ongoing basis, as it would be desirable to produce results in real-time. In the next section, the real-time nature of the three models is discussed.

7.2.3 Real-Time

One of the main aims of continuous auditing is to provide assurance information in as close to real-time as possible. In all three of the models, providing real-time assurance is the aim.

Real-time accounting requires real-time auditing to provide the desired continuous assurance about the quality of the data. While this has long been desirable, one of the prohibiting factors in publishing real-time reports in the past was cost. Thus, reports were published periodically. It was too costly to obtain the information required to produce the reports. It is now possible to produce real-time, standardised financial information, online. In the model by Rezaee et al, data marts are described as an auditing approach which will facilitate real-time analysis and reporting in a real-time accounting environment (Rezaee et al., 2002).

Onions (2003) argues that since business is carried out in real-time, auditing of transactions should also be carried out in real-time. Analysing data needs to be done in a dynamic way, close to simultaneously with the entry of the transaction. The data must be policed from entry through all the sub-ledgers to the final posting in the general ledger. This is why the adaptation of CAATTs to run real-time is suggested. Communication with OLSAC is also real-time, as opposed to annual. While most of the analysis is performed in parallel with the operation of the audited systems, certain aspects of this system would not be able to run as the transaction is entered. Analysis by expert systems, while not running when the transaction is entered, is however continuous. It is performed once the transaction has been accumulated with other transactions in a data warehouse.

Woodroof and Searcy (2001) claim that many of the innate flaws which negatively affect the quality of information, and thus the quality of decision-making, have to do with timeliness and relevance. Evergreen reports aim to give an instantaneous report over the Internet. Evergreen reports are discussed in more detail in the next section.

Although the three models each present a different methodology, they all aim to accomplish real-time reporting and assurance. Each of the models relies on different core technologies to achieve their aims. Further differences exist in the nature of the reports and alerts produced by each model. These differences will be elaborated on in the next section.

7.2.4 Reporting Method

There are various methods of reporting used in different continuous auditing systems. Generally, the results of tests performed by analytic procedures (for example expert systems and digital agents) are used to create reports and alerts, which are sent to auditors by encrypted email. The alerts may also be sent via Virtual Private Networks (VPNs). A grading system for alerts, showing the possible impact of the anomaly, may be important to the auditors. A grading system is a possible method of sending alerts only to parties to whom they are relevant; if too many irrelevant alerts are received, the credibility of the system my be questioned (Onions, 2003). Onions also suggests that instead of grouping alerts simplistically (e.g., Red, Amber, Green), a cumulative scoring system to give weighting to alerts could be developed. Woodroof and Searcy's model also uses a grading system for reports, which are issued relating to one of three levels of assurance.

The reporting system of the Woodroof and Searcy model differs from that of Onions. The Woodroof and Searcy model differs from most continuous auditing systems in that it uses a 'pull' approach as opposed to the usual 'push' reporting approach. In the push approach, used for example in the Onions model, reports and alerts are generated by the system, and when certain conditions are met, these are automatically sent to the user (auditor). In contrast, the report is requested by the user in the pull approach. The implementation of the pull reporting approach is accomplished through use of a Web interface. This interface is how the user interacts with a Web application, developed in Microsoft's Active Server Pages (ASP) and JavaScript, and

hosted by Web Servers, which utilize such software as Allaire's Cold Fusion, Lotus Domino, Sybase Enterprise Server, Oracle, Netscape, Bluestone's Sapphire/Web and Net Dynamics. *Evergreen reports* are the method used to present the reported information. As mentioned in Chapter Six, evergreen reports are reports which are generated on demand, usually viewed through a Web site, and are dynamically dated and time stamped when the user visits the site. Other alerts created in the system by digital agents are sent to the users via email.

In order to produce reliable, credible reports, the underlying information must be reliable and credible. The integrity of the underlying data must be protected from when it is acquired, through processing of transactions, to the production of reports, as discussed in Chapter Five. In continuous auditing systems, accessing the required data without compromising the integrity can be a complex procedure. Different methodologies exist for acquiring and formatting data so that an audit can successfully occur. In the next section, some of the solutions from the compared models will be explained.

7.2.5 Data Acquisition and Solving the Data Format Problem

Before any auditing can take place, the required data needs to be acquired from the production databases. Only once the data has been extracted from these production databases can it be used within CAATTs, such as in off-the-shelf database applications (e.g. Microsoft Access), spreadsheets (e.g. Excel) or data analysis applications (ACL, IDEA and Monarch etc) (Champlain, 2003).

In order to acquire data, it must be extracted, transformed and loaded (the ETL process). Transactions from a variety of sources are extracted. Not all records or fields may be required, digital agents and stored database procedures could be used to extract only the necessary data (Rezaee et al., 2002). Before continuous auditing, the data extracted by a report writer, created by the data owner in conjunction with the auditor, is sent to the auditor's workstation via a network. The auditor can then use the aforementioned CAATTs to analyse data. In a continuous auditing system, the data is extracted to data marts where standardised audit tests can automatically gather evidence and generate exception reports. The data mart also holds metadata documenting source transactions and the ETL process. This metadata details the data

extraction and transformation processes and the audit tests which were applied to the data. Stored data extraction information includes which source tables were used, and which columns were selected. The information stored about the data transformation process regards the appending, renaming, labelling and sorting of the audit data.

To facilitate the creation of data marts a capable Database Management System (DBMS) would at least be required, if not a specialized data warehousing solution. Data marts and data warehouses were alluded to in Chapter Six. Within the context of a traditional IS audit, a data warehouse is described as a sizable database which provides a way to access information from two or more different systems or sources. They are advantageous as data warehouses reduce the need for multiple report writers and data extraction programs. Multiple or complex report writers may be heavy on processing power, and therefore are to be avoided, as they can slow down the production system (Champlain, 2003).

Data warehouses are designed to support management decision-making. They are further defined as an integrated, subject-oriented, time-variant, non-volatile database (Rob & Coronel, 2002). A data mart is usually smaller than a data warehouse and relates to a single subject only. It can be a subset of a data warehouse (Rob et al., 2002). Data warehouses are not only useful for examining transactions, but also allow the auditor to search for potential internal control weaknesses (Champlain, 2003). Examples of specialised data warehouse software include; Oracle Warehouse Builder, Sagent, IBM Red Brick and Kalido Software. These data warehousing solutions are currently used mostly in the field of Customer Relationship Management (CRM) (Champlain, 2003; Rochnik, 2006).

One of the prominent obstacles encountered in creating an efficient and effective continuous auditing system is that of easily accessing and retrieving data from various systems and platforms. Data is often desired from software on employee workstations as well as enterprise computing platforms/ERP systems, such as SAP R/3, BaaN, PeopleSoft, JD Edwards, Sage, Pegasus, Great Plains and DBMSs such as Oracle. This data are often stored with diverse file types and record formats such as IMS, VSAM, ASCII, MBD, CSV, XLS and TXT. This processes of accessing the data, as well as data from legacy systems, and converting it into a usable form is known as

standardisation. This process is complex and can be expensive. There is also the risk of losing the integrity of the data when standardising it due to introducing errors and duplicating records. This has been called the *data format problem* within this dissertation.

Data standardisation necessitates the development of a series of standards for storing data in the audit data warehouse (Rezaee et al., 2002). Standardisation may be done either on the audit server or in the source application, depending on the cost of processing. For example, the cost of processing on mainframes is more expensive. The data can then be aggregated in data marts or data mines.

Only the auditors, business unit managers and corporate security officers need access to the audit data marts, as they will perform audit testing and exception reporting duties from their workstations. The data mart or data mine allows them to perform their audit duties without the need for involving themselves with the complex data acquisition and the ETL process.

Data mines are expensive and normally only larger organisations can afford them. Data marts are often used by smaller organisations, or in larger organisations for one specific focus area, for example, human resources, accounting, etc. (Rezaee et al., 2001). The solution suggested by Rezaee et al is using data marts. Smaller organisations could also make use of XCAL, as suggested by Onions (2001). Woodroof and Searcy do not interface with legacy systems and do not address this issue directly.

7.3 Conclusion

In this chapter, the three models described in Chapter Six were evaluated and compared. A table was presented, which summarized how the key aspects of a continuous auditing system are addressed by each model. The key differences between how each of the models addresses these aspects was then highlighted. The comparison firstly showed that when evaluating the accuracy of transactions, the presence of fraud and error within transactions needs to be detected by technology. The technologies which could be most useful include CAATTs and standardised tests, expert systems and digital agents. Secondly, the comparison revealed that when

verifying the reliability of the system of internal controls; CAATTs, such as ITF, parallel simulation, concurrent processing audit modules and CIS, prove to be useful. The third criterion in the table was whether the models functioned in real-time. All the models aimed to give results in as close as real-time as possible.

When the reporting methods used by each model were compared, the biggest difference was that the Woodroof and Searcy model used a Web-centric pull methodology, while the other two used a push methodology to send alerts via secured networks.

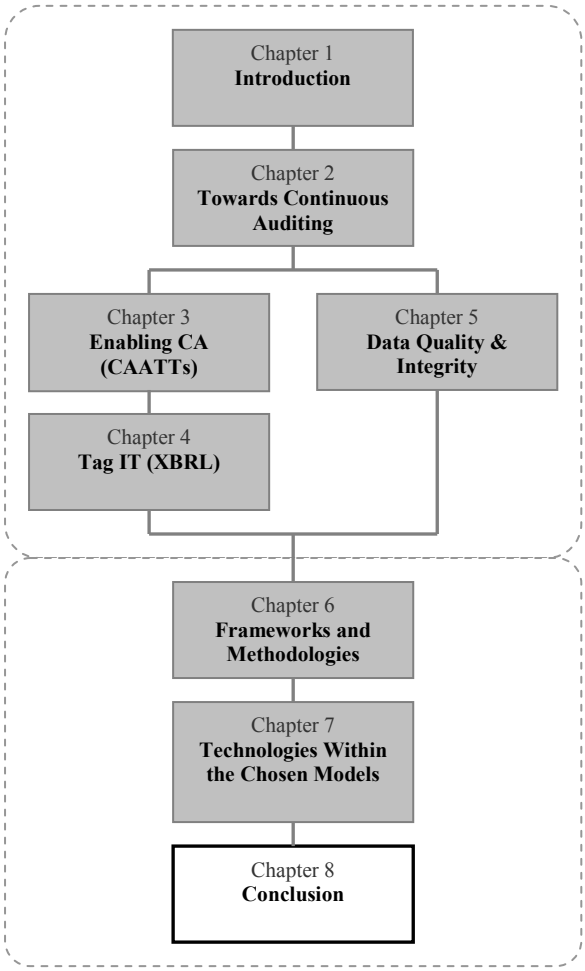
An important part of creating a successful continuous auditing system is acquiring data and overcoming data format problems. In the model proposed by Rezaee et al, data marts were proposed as a solution, while Onions proposed the development and use of a XBRL-based language, called XCAL. This problem had minimal impact in the Woodroof and Searcy model, as most external data was acquired through digital agents, and there was no interface with legacy systems.

Each of the three models had a slightly different aim. The Rezaee et al model presented a generic overview of a continuous auditing system. The model illustrated that data marts can be used to assist in solving data access issues. The model proposed by Onions is more specific than that of Rezaee et al, as the solution covers more specific technologies, in greater depth. This model also presents XCAL as a unique feature, to solve data format and accessibility problems and facilitate the use of expert systems. The Woodroof and Searcy model is the most specific model, as it discusses the application of continuous auditing to a single business domain. It is also far more Web-centric than the others. It relies on digital agents to communicate between the parties, and it has a Web-based reporting model which produces reports on request.

A comprehensive solution could take the best aspects from each of these models. These models also prove the effective use of existing technologies within the context of continuous auditing. Solutions to the data format problem and methods of effective data acquisition were also described.

Chapter 8

Conclusion



Chapter 8:

8.1 Summary of Chapters

8.2 Research Objectives

8.3 Limitations and Further Research

8.4 Epilogue

8.1 Summary of Chapters

The first five chapters provided a background view to continuous auditing. Continuous auditing was firstly defined and explained and then the environment in which continuous auditing developed was examined. This includes the developments in the technological environments as well as information quality and integrity.

Firstly continuous auditing was defined. The focus in this dissertation was on the ability of continuous auditing to provide real-time reporting, this would in turn facilitate better decision-making. Other important discoveries regarding continuous auditing, as far as this dissertation is concerned, were that continuous auditing is a tool useful to both internal and external auditors, and that the testing of controls must

be done simultaneously with substantive tests of transactions (Helms et al., 1999; Rezaee et al., 2001).

Much of the desire for continuous assurances, which can be provided by continuous auditing, was driven by legislation, such as the Sarbanes-Oxley Act. This legislation came about after several lapses in corporate governance were made public by debacles such as Enron. The focus of audit engagements also changed. It shifted from manual detection of misstatements and fraud to that of technology-based prevention (Bierstaker et al., 2001; Rezaee et al., 2001).

At the time that the above debacles occurred many of the technologies which are required for continuous auditing became available. This included more powerful processors, which are required in order to perform real-time processing of a large number of transactions. Disk Mirroring, for example RAID (redundant array of independent disks) allowed for more reliable mass storage of data. Vast amounts of cheap storage also became a reality, and this allowed the storage of the large amounts of data which are required by large databases and data marts, and also allowed the storage of archived transactions, possibly for protracted periods of time, for future reference. Faster communication also became a reality; it is one of the most essential requirements for continuous auditing. Real-time reporting is not possible if the required information cannot be efficiently and promptly accessed. Faster information exchange was enabled by increased network bandwidth and specifically the ability to communicate over extended networks, such as the Internet. Finally, more secure systems completed the picture. Secure systems are required as the stored, transmitted and processed data, especially financial or performance information, may be sensitive. Strong encryption algorithms are also needed for communication. Security is also important as security affects reliability, and a reliable system is one of the essential requirements for continuous auditing (Srinivas, 2006a).

Increased systems integration became widespread and was typified by the use of ERP software. EDI and EFT had also changed the technological landscape of business, by blurring the communication boundaries between partnered organisations, and this also reduced the paper-trail on which traditional audits were based. Documents such as purchase orders, invoices and cheques, now only existed in electronic format.

Electronic Data Processing, which had grown from infancy in the 1970s had to develop to assist the auditors in this new electronic business environment.

Chapter Three introduced the three essential requirements for continuous auditing (Shields, 1998). These requirements were documented in a report by CICA (CICA/AICPA, 1999). The first requirement is that the information to be audited has to be produced by a reliable system. The second is that the continuous auditing must be part of a highly automated process, and that tools must be imbedded within the client's system. The final requirement is that there has to be a fast, accurate and secure communication channel between the auditor and the client's systems.

The requirement for a communication channel was easily met during the 1990s when network bandwidth increased and specifically the ability to communicate over extended networks, such as the Internet. Extranets between organisations became commonplace as Virtual Private Networks (VPNs) and strong encryption algorithms developed. Faster information exchange acted as a driver to real-time reporting as it is not possible if the required information cannot be efficiently and promptly accessed.

The second requirement could possibly be addressed through the use of certain CAATTs. This includes tools such as embedded audit modules and GAS. CAATTs, in general, are essential to auditors, as they can be used to verify that a real-time accounting system is producing reliable and credible financial information (Helms et al., 1999; Rezaee et al., 2001). CAATTs can then be grouped into two broader categories. Firstly CAATTs used for *analysing transactions*, and secondly those used for *testing internal controls and assessing risk*.

The CAATTs most suited to analyzing transactions are GAS, EAMs and Artificial Intelligence related technologies. CAATTs which can be used for testing internal controls include: using test data, integrated test facilities (ITFs), embedded audit modules (EAMs), parallel simulation and concurrent processing (Cerullo et al., 2003). Neural networks may also be of some use in testing internal controls.

The final requirement is for reliable systems. To this end Chapter Four discussed how XML-based technologies, such as XBRL can enable better reliability. XBRL holds

many advantages for continuous auditing. Most notably, it creates a standardised data format. This eliminates much of the need to re-capture data between systems, this in turn helps to reduce errors and saves time. An additional advantage is that metadata is conveyed along with the data. An added advantage is that the use of a standardised data format facilitates the use of web-enabled audit programs, which are often integral in continuous auditing systems (Coderre, 2004).

Many of the shortfalls of XBRL can be addressed by adding XARL. XARL focuses on addressing the security aspects of XBRL, specifically the correctness and completeness of data. Other XML-based technologies introduced in Chapter Four; differ from XBRL in that they are transaction oriented, unlike XBRL which is a reporting language.

Chapter Five introduced the concepts of Integrity and Quality. These concepts are important to an understanding of continuous auditing, as the auditor is required to gather persuasive evidence regarding the quality and integrity of the system in producing reliable and credible information (Rezaee et al., 2002). Verifying the integrity of data before fulfilling the audit objectives is essential, so that the conclusions in the auditor's reports can be shown to be based on accurate and reliable data (Wessmiller, 2002). This is in line with regulations such as Sarbanes-Oxley.

The last two chapters proposed ways to integrate and adapt the technologies essential to continuous auditing. To discover which technologies are essential Chapter Six compared and evaluated three of the most influential continuous auditing models, discussed in the surveyed literature.

The models differed in focus or aim. Rezaee et al's model was the most generic and discussed continuous auditing methodology as well as addressing the problem of data standardisation. Onions focuses more on preserving the integrity of data and overcoming the problem of data formats, for which XCAL is suggested. Woodroof and Searcy aimed to develop a working prototype of a continuous-auditing system, however the scope of the model was limited to a single business situation – debt-covenant compliance.

The steps involved in each model and the component parts of each model were discussed, and some of the important technological components used within each of the models were highlighted.

Chapter Seven built on the foundation laid by Chapter Six by comparing the models and highlighting commonalities between the technologies within the models. In evaluating the models, the way in which accuracy and reliability are validated, were compared. *Accuracy* refers to how fraud and error in transactions are detected and how possible material misstatements in financial records are detected, while *Reliability* is how confidentiality, integrity and availability of internal controls are examined. The other aspects which were compared included the real-time nature of each model, the reporting method used and how the problem of differing data formats is addressed.

The data format problem was defined and discussed. One of the prominent obstacles encountered in creating an efficient and effective continuous auditing system is that of easily accessing and retrieving data from various systems and platforms. There is a need to access data from software on employee workstations as well as enterprise computing platforms/ERP systems, such as SAP R/3, BaaN, peopleSoft, JD Edwards, Sage, Pegasus, Great Plains and DBMSs such as Oracle. However, this data is often stored with diverse file types and record formats such as IMS, VSAM, ASCII, MBD, CSV, XLS and TXT. One method of accessing the data, as well as data from legacy systems, and converting it into a usable form is standardisation, however it is complex and can be expensive. There is also the risk of losing the integrity of the data when standardising it due to introducing errors and duplicating records. Using a standardised data format, such as XBRL, will go a long way towards eliminating this problem in the future. Agent technology such as FRAANK could assist in accessing legacy data, by translating it into XBRL.

Other essential technologies include data warehouse and DBMS as well as data mines or data marts. The solution suggested by Rezaee et al is using data marts, as data warehouses can be expensive. Smaller organisations could make use of XCAL, as suggested by Onions (2001), while Woodroof and Searcy do not interface with legacy systems and so do not address this issue directly.

Furthermore, this chapter looked at how accuracy and reliability were examined by the model. Examining for the accuracy of transactions involves establishing the existence of fraud and error. In doing this, CAATTs can be employed as an invaluable tool; some CAATTs would however need to be modified so that they are no longer batch tools, but run in real-time. Data Query Modules (DQMs) can also be of assistance. These tools would be employed in both substantive testing and performing analytical procedures. Digital agents and CIS are particularly useful for selecting transactions to be examined further by the auditor and then setting these aside. Once the data has been standardised and stored, it can then be examined for the cumulative effects of groups of transactions, as proposed by the Onions model (2003). Expert systems may provide the best way to do this, although in the future, self-learning technologies like Neural Networks could also be employed.

Verifying the reliability of the system involves examining the internal control structure. It needs to provide confidentiality, integrity and availability to the data within the system. CAATTs such as parallel simulation and integrated test facilities (ITFs), can be used to achieve this. Concurrent audit techniques include SCARF (Systems Control and Review Facility) and the snapshot approach. Other methods for checking that the internal control structure is adequate in ensuring the integrity and reliability of the audit data involve parsing keystrokes, implementing Operating System security and passwords protection, creating adequate audit logs, using web services and making sure that guidelines and best practices such as SYSTRUST, WEBTRUST, eSAC, COSO and COBIT have been adhered to.

The real-time nature of performing the audit is stressed in all three of the models. To this end it is proposed that the auditing of transactions begins as soon as they are entered (Onions, 2003). This monitoring and reporting on data should continue throughout the life-span of the data. Another aspect to the real-time nature of continuous auditing is that communication with outside auditors and verification authorities is done in real-time when a query arises, this is made possible by fast and secure communication channels, such as VPNs.

The reporting methods used for each model were also discussed. A common feature which is required is a grading or weighting system, to indicate the severity/gravity of the report. The concept of evergreen reports, as proposed by Woodroof and Searcy, was introduced. This is a new type of on demand report, which always appears current. The main difference between the Woodroof and Searcy model, as compared to the other two models, is that it uses a pull approach, whereby the user can request a specific report at anytime. Pull reporting may be an important mechanism in facilitating the much desired improvements in the data which support decision-making. The reports are produced when the data is relevant, as they are requested by the user. The data are also always current and will be produced rapidly, unlike in traditional auditing where annual reports are produced long after the events have occurred.

8.2 Research Objectives

The primary objective of this dissertation was to investigate and identify technologies which assist in providing continuous assurances by supporting continuous auditing. This primary research objective was the focus of Chapter Seven, but in order to fully address it, three secondary objectives were first discussed. These are:

1) Data Format Problem

This was addressed in Chapter Seven, and is described by a comparison which dealt with how each model managed the acquisition of data in different formats. The problem of needing data which were in several different formats was entitled the *data format problem*. This problem is aggravated when data from legacy systems are required. This problem addressed the first of the secondary research objectives, and standardisation of the data is proposed as a solution. The data needs to be converted to a standard format and stored in audit data warehouses, from where all aggregated data can be accessed. Data mines can then be used by auditors to sort data, but as these solutions may be expensive, smaller organisations may use data marts or XCAL. Many of these problems may be of less concern in the future, due to standardised reporting methods with metadata, such as XBRL, which was discussed in Chapter Four. When archived reports are in XBRL format, they can quickly be read and compared to current reports. This will assist in bringing audits even closer to real-time, as data standardisation can

be time-consuming. The accessibility of audit data will be greatly improved and data-corruption risks are also reduced if the need for standardization is diminished.

2) Data integrity and quality

The aforementioned data-corruption risks of standardisation relate to the concepts of integrity and reliability of data, thus, Chapter Five introduced and discussed these concepts and offered definitions of data integrity and quality.

Clarifying the concepts of data integrity and quality was achieved by means of a topical literature survey, these terms were defined and their component attributes were discussed. It was established that continuous auditing requires reliable data, thus the integrity and quality aspects of data are important concepts.

3) Accuracy and Reliability of the data, information and system

Threads relating to these concepts are scattered throughout this research project. Tools which assist in maintaining the accuracy and reliability of data, information and the system were introduced and discussed. Chapter Three introduced CAATTs as enablers of continuous auditing and explained their dual role. The first role is that of analyzing transactions, this relates to establishing the accuracy of transactions. The second is in testing internal controls and assessing risk, this relates to establishing the reliability of internal controls.

Meeting the Primary Objective:

In order to meet the primary objective, the technologies proposed within the continuous auditing models, discussed in Chapter Six, were categorized according to purpose. These were categorised according to accuracy, specifically how transactions are handled, and reliability, which focused on internal control examination.

In Chapter Seven a tabulated summary compared and contrasted the three models. This, besides addressing accuracy and reliability, also compared the reporting method and real-time nature of the models. In Chapter Seven it was discovered that a comprehensive continuous auditing solution would need to use the best aspects from

the compared continuous auditing models. The objectives of this research project were met primarily through a critical analysis of the surveyed literature.

8.3 Limitations and Further Research

A limitation of this research is that continuous auditing may report on many aspects of an organisation, and while financial reports are relatively similar between organisations, the requirements for internal controls may differ. For example, internal control reporting in financial institutions may need to be especially comprehensive. Business sector-specific legislation may also exist and would require compliance assessment. For example, in the Health Sector legislation emphasises data confidentiality and integrity. A continuous audit may include reports on such aspects as well as Key Performance Indicators, which differ between organisations. It would therefore be prudent in future research to limit the scope to a particular organisation, to avoid a "Cinderella's shoe" situation, where a solution is created but cannot be tested because of difficulty locating an organisation for which the solution is fully applicable.

This research could perhaps benefit from the design of a prototype continuous auditing application, applicable to industry. However, because each organisation has differing needs and requirements and may have a different combination of systems architectures and software the prototype may still have limited applicability to many organisations.

Further research may focus on security, and in particular could include closer investigation of the available methods for securing XML-based information transmitted over the Internet.

8.4 Epilogue

A comprehensive solution could take the best aspects of each of these models. These models also prove the effective use of existing technologies within the context of continuous auditing. A comprehensive technical model for continuous auditing does not yet exist, this may well be due to the fact that each company and situation will require a unique blend of technologies. Research taking a pragmatic view of

continuous auditing is required to complete the picture which was outlined by the theoretical models suggested by the auditing fraternity.

References

Alles, M., Kogan, A., & Varsarhelyi, M. A. (2002). Feasibility and Economics of Continuous Assurance. *Auditing: A journal of Practice and Theory* (March), 125-138.

Alles, M., Kogan, A., & Vasarhelyi, M. (2003). Black Box Logging and Tertiary Monitoring of Continuous Assurance Systems. *Information Systems Control Journal, 1*.

Alles, M., Kogan, A., & Vasarhelyi, M. (2004). *Real Time Reporting and Assurance: Has Its Time Come?* Retrieved 12 May, 2005, from http://raw.rutgers.edu/continuousauditing/Real_Time_Reporting_-_ICFAI1.doc

American Institute of Certified Public Accountants (AICPA). (1996). *Amendment to Statement on Auditing Standards No. 31, Evidential Matter: SAS 80*.

Bierstaker, J. L., Burnaby, P., & Thibodeau, J. (2001). The Impact of Information Technology on the Audit Process: An Assessment of the State of the Art and Implications for the Future. *Managerial Auditing Journal, 16*(3), 159-164.

Boritz, J. (2004). *Managing Enterprise Information Integrity - Security, Control and Audit Issues*. USA: IT Governance Institute.

Boritz, J. E. (2005). IS Practitioners' Views on Core Concepts of Information Integrity. *International Journal of Accounting Information Systems, 6*, 260-279.

Boritz, J. E., & No, W. G. (2003). Assurance reporting for XBRL: XARL. . In J. Roohani (Ed.), *Trust and data assurances in capital markets: the role of technology solutions* (pp. 17-31): PricewaterhouseCoopers LLP.

Boritz, J. E., & No, W. G. (2004). Business Reporting with XML: XBRL (Extensible Business Reporting Language). In H. Bidgoli (Ed.), *Encyclopoedia of the Internet* (Vol. 2): John Wiley.

Braun, R. L., & Davis, H. E. (2003). Computer-assisted audit tools and techniques: analysis and perspectives. *Managerial Auditing Journal, 18*(9), 725-731.

Canadian Institute of Chartered Accountants. (1998). *Information Technology Control Guidelines*. Toronto (ON).

Cerullo, M. J., & Cerullo, M. V. (2006). Using Neural Network Software as a Forensic Accounting Tool. *Information Systems Control Journal*.

Cerullo, M. V., & Cerullo, M. J. (2003). Impact of SAS no 94 on Computer Aided Audit Techniques. *Information Systems Control Journal, 1*.

Champlain, J. (2003). *Auditing Information Systems* (2nd ed.). Hoboken, New Jersey: John Wiley & Sons.

CICA/AICPA. (1999). Continuous Auditing: Research Report: Canadian Institute of Chartered Accountants.

Coderre, D. (2001). *CAATTS and other BEASTS*. Vancouver, Canada: ACL Institute.

Coderre, D. (2004, August). Are You Ready for XBRL? *Internal Auditor,* 26-28.

Cohen, E. E. (2002). Data Level Assurance: Bringing Data into to [sic] Continuous Audit Using XML Derivatives, *Fifth Continuous Assurance Symposium*. Rutgers Business School, Newark, NJ.

Collis, J., & Hussy, R. (2003). *Business Research: A Practical Guide for Undergraduate students* (2nd ed.). Basingstoke: Palgrave Macmillan.

Dalal, C. (1999). Using an Expert System in an Audit: A Case Study of Fraud Detection. *IT Audit, 2* (May 15).

Dalal, C. (2000). Advanced use of audit software in audit and fraud detection - audit software: an indispensible tool in the new millennium. *IT Audit, 3* (February 1).

Debreceny, R., & Gray, G. L. (2001). The production and use of semantically rich accounting reports on the Internet: XML and XBRL. *International Journal of Accounting Information Systems*(2), 47-74.

Finnigan, P. (2003). *Introduction to simple oracle auditing*. Retrieved 10 April, 2006, from http://www.securityfocus.com/print/infocus/1689

Flowerday, S., Blundell, A. W., & R. von Solms. (2006). Continuous auditing technologies and models: A discussion. *Computers & Security, 25*(5), 325-331.

Flowerday, S., & R. von Solms. (2005a). Continuous Auditing: Verifying Information Integrity and Providing Assurances for Financial Reports. *Computer Fraud and Security, 7*, 12-16.

Flowerday, S., & R. von Solms. (2005b). Real-time information integrity = system integrity + data integrity + continuous assurances. *Computers and Security, 24*, 604-613.

Franklin, S., & Graesser, A. (1996). *Is it an Agent, or just a Program?: A Taxonomy for Autononmous Agents*. Paper presented at the Third International Workshop on Agent Theories, Architectures and Languages, Springer-Verlag.

Garthwaite, C. (2000). The Language of Risk: Why the Future of Risk Reporting is Spelled XBRL. *Balance Sheet, 8*(4), 18-20.

Glover, S., & Romney, M. (1997). Software - 20 Hot Trends. *The Internal Auditor, 54*, 45-48.

Groomer, S. M., & Murthy, U. S. (1989). Continuous Auditing of Database Applications: An Embedded Audit Module Approach. *Journal of Information Systems, (Spring) 3*(2), 53-69.

Hannon, N. (2005, April). XBRL Fundamentals. *Strategic Finance,* 57-58.

Helms, G. L., & Mancino, J. M. (1999). *The CPA & the Computer: Information Technology Issues for the Attest, Audit, and Assurance Services Functions*. Retrieved 3 May, 2005, from http://www.nysscpa.org/cpajournal/1999/0599/departments/cpac.html

ISACA Standards Board. (2002). Continuous Auditing: Is it Fantasy or Reality? *Information Systems Control Journal, 5*.

ISO/IEC 17799. (2000). *Information Technology - Security Techniques - Code of practice for information security management. International Organization for Standards*, from http://www.iso.org/iso/en/ISOOnline.frontpage.

Kogan, A., Nelson, K., Srivastava, R., Vasarhelyi, M., & Bovee, M. (1998). *Design and Applications of an Intelligent Financial Reporting and Auditing Agent with Net Knowledge*. Retrieved 10 May, 2005, from https://kuscholarworks.ku.edu/dspace/bitstream/1808/141/1/srivastava.pdf

Kogan, A., Sudit, E. F., & Varsarhelyi, M. A. (1999). Continuous Online Auditing: A Program of Research. *Journal of Information Systems, 13*(2).

Kogan, A., Sudit, F., & Vasahelyi, M. (2000). *Some Auditing Implications of Internet Technology*. Retrieved 16 February, 2006, from http://www.rutgers.edu/accounting/raw/miklos/tcon3

Koskivaara, E. (2003). *Artificial Neural Networks in Auditing: State of the Art*. Retrieved Oct 31, 2005, from www.tucs.fi/publications/attachment.php?fname=TR509.pdf

Krell, E. (2004). *"Continuous" Will be Key to Compliance*. Retrieved 20 June, 2005, from http://www.bfmag.com/magazine/archives/article.html?articleID=14337&Print=Y

McCollum, T., & Salierno, D. (2003). Choosing the Right Tools. *Internal Auditor, August*, 32-43.

Mookhey, K. K. (2004). *Auditing Oracle Security*. Retrieved 16 April, 2005, from http://www.theiia.org/itaudit/index.cfm?fuseaction=print&fid=5509

Murthy, U. S., & Groomer, S. M. (2004). A Continuous Auditing Web Services Model for XML-based Accounting Systems. *International Journal of Accounting Information Systems, 5*, 139-163.

National Security Telecommunications and Information Systems Committee. *National Training Standard for Information Security Professionals NSTISSI No 4011*, from www.nstissc.gov/html/library.html

Naumann, J. (2004). *Tap Into XBRL's Power the Easy Way*. Retrieved 4 August 2005, from http://www.aicpa.org/pubs/jofa/may2004/naumann.htm

Onions, R. L. (2003). *Towards a paradigm for continuous auditing*. Retrieved 1 April, 2005, from http://www.auditsoftware.net/community/how/run/tools/Towards%20a%20Paradigm%20for%20continuous%20Auditin1.doc

Pinkster, R. (2003). XBRL awareness in auditing: a sleeping giant? *Managerial Auditing Journal, 18*(9), 732-736.

Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data Quality Assessment. *Communications of the ACM, 45*(4), 211-218.

Ramamoorti, S., & Weidenmier, M. L. (2004). *Research Opportunities in Internal Auditing: Chapter 9 The Pervasive Impact of Information Technology on Internal Auditing (Supplemental Chapter)*. Alamonte Springs, Florida: The Institute of Internal Auditors Research Foundation (IIARF).

Redman, T. C. (1998). The Impact of Poor Data Quality on the Typical Enterprise. *Communications of the ACM, 41*(2), 79-81.

Rezaee, Z., Elam, R., & Sharbatoghlie, A. (2001). Continuous Auditing: the Audit of the Future. *Managerial Auditing Journal, 13*(3), 150-158.

Rezaee, Z., & Reinstein, A. (1998). The Impact of Emerging Information Technology on Auditing. *Managerial Accounting Journal, 13*(8), 465-471.

Rezaee, Z., Sharbatoghlie, A., Elam, R., & McMickle, P. (2002). Continuous Auditing: Building Automated Auditing Capacity. *Auditing: A journal of Practice and Theory, 21*(1), 147-163.

Rob, P., & Coronel, C. (2002). *Database Systems - Design, Implementation and Management* (5th ed.): Thompson Learning.

Rochnik, N. (2006). *Oracle Warehouse Builder 10gR2 Transforming Data into Quality Information*. Retrieved 5 September, 2006, from www.oracle.com

Shields, G. (1998). Non-stop Auditing. *CAMagazine* (September 1998), 39-40.

Snedaker, S. (2006). *IT Security Project Management Handbook*. Rockland, MA: Syngress Publishing.

South African Institute of Chartered Accountants. (2003). *SAICA Handbook - Auditing* (2003/2004 ed. Vol. 2).

Srinivas, S. (2004). Road Map to XBRL Adoption as a New Reporting Model. *Information Systems Control Journal, 1*.

Srinivas, S. (2006a). Continuous Auditing through Leveraging Technology. *Information Systems Control Journal*.

Srinivas, S. (2006b). *Continuous Auditing Through Leveraging Technology*. Retrieved 1 May, 2006, from www.isaca.org

Strong, D. M., Lee, Y. W., & Wang, R. Y. (1997). Data Quality in Context. *Communications of the ACM, 40*(5), 103-110.

Taylor, P. (2005). The perils of systems-based fraud. *IT Audit, 8* (January 15).

The Canadian Institute for Chartered Accountants. (2002). Audit and Control Implications of XBRL. Toronto Canada.

The IT Governance Institute. (2005). (COBIT) Control Objectives for Information and related Technology, (4th Ed.). *USA*.

Varsarhelyi, M. A., Kogan, A., & Alles, M. G. (2002). Would Continuous Auditing have Prevented the Enron Mess? *CPA Journal, 72*(7).

Vasarhelyi, M. A. (2002). *Concepts in Continuous Assurance*. Retrieved March, 2005, from http://raw.rutgers.edu/continuousauditing/conceptsincontinuousassurance13final.doc

Vasarhelyi, M. A., & Halper, F. B. (1991). The Continuous Audit of Online Systems. *Auditing: A Journal of Practice and Theory, 10*(1).

Wand, Y., & Wang, R. Y. (1996). Anchoring Data Quality Dimensions in Ontological Foundations. *Communications of the ACM, 39*(11), 86-95.

Wessmiller, R. (2002). *Facing the Data Integrity Challenge*. Retrieved 20 April, 2005, from http://www.theiia.org/itaudit/index.cfm?fuseaction=print&fid=440

Whitman, M. E., & Mattord, H. J. (2003). *Principles of Information Security*: Thompson Course Technology.

Willis, M. (2005, March). XBRL and Data Standardization: Transforming the Way CPAs Work. *Journal of Accountancy,* 80-81.

Willis, M., & Hannon, N. J. (2005, July). Combating Everyday Data Problems with XBRL. *Strategic Finance,* 57-59.

Woodroof, J., & Searcy, D. (2001). Continuous Audit: Model Development and Implementation within a Debt Covenant Compliance Domain. *International Journal of Accounting Information Systems, 2*, 169-191.

PART III – Appendices

Appendix A

Published Article

The following article was published in the journal *Computers and Security*, volume 25, issue 5.

available at www.sciencedirect.com

SCIENCE DIRECT®

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Continuous auditing technologies and models: A discussion

*S. Flowerday, A.W. Blundell, R. Von Solms**

Centre for Information Security Studies, Nelson Mandela Metropolitan University, Port Elizabeth, South Africa

ARTICLE INFO

Article history:
Received 13 June 2006
Revised 13 June 2006
Accepted 13 June 2006

Keywords:
Continuous auditing
Real-time assurances
Information integrity
Internal controls
Technology-based prevention

ABSTRACT

In the age of real-time accounting and real-time communication current audit practices, while effective, often provide audit results long after fraud and/or errors have occurred. Real-time assurances can assist in preventing intentional or unintentional errors. This can best be achieved through continuous auditing which relies heavily on technology. These technologies are embedded within and are crucial to continuous auditing models.

1. Introduction

In today's fast paced business world, real-time information systems facilitate real-time accounting systems and real-time communication between entities. Current audit practices, while proving adequate, take too long to provide assurances. Current audit practices also uncover intentional and unintentional errors, however, only after it has possibly had a detrimental effect on the organisation. A method of prevention by detecting these errors early is desirable. It is time to provide real-time assurances to decision makers.

Real-time assurances can only be provided by continuous auditing technologies. This paper discusses a range of audit technologies within three continuous auditing models. Each of these models has a different focus and makes use of different technologies. The available technologies, tools and techniques used in continuous auditing, will be interrogated.

The aim of this paper is to reach a conclusion about how a generic, yet comprehensive continuous auditing system could make use of the available tools, techniques and technologies for testing internal control and performing tests on transactions. To achieve this aim, after discussing the history and definition of continuous auditing the reasons for using continuous auditing systems will be explored. Once this is clarified, the tools and techniques necessary to implement continuous auditing are discussed, these are then placed into context by discussing three prominent continuous auditing models. The three models are then compared in tabulated form, after which possible future technologies are suggested for addressing internal control issues and testing transactions within a continuous auditing system.

2. Definition and history of continuous auditing

There are several differing ideas of what continuous auditing (CA) systems are, and how they work. Each of the models

* *Corresponding author.*
E-mail addresses: sflowerday@telkomsa.net (S. Flowerday), ablundell@nmmu.ac.za (A.W. Blundell), rossouw@nmmu.ac.za (R. Von Solms).
0167-4048/$ – see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2006.06.004

(discussed later) has their own definition, differing slightly. The most widely accepted definition though, is one released in 1999 and reads as follows: ''*a methodology that enables independent auditors to provide written assurance on a subject matter using a series of auditors' reports issued simultaneously with, or a short time after, the occurrence of events underlying the subject matter*'' (CICA/AICPA, 1999). This is the definition used for the purpose of describing continuous auditing throughout this paper.

In the early 1990s, the business environment went through a series of substantial changes. The ''*Electronization*'' of business and the proliferation of e-business lead to paperless accounting systems (Bierstaker et al., 2001; Vasarhelyi, 2002). This move towards technologies such as Electronic Data Interchange (EDI) and Electronic File Transfer (EFT) caused the evaporation (or disappearance) of the traditional audit trail. Auditors could no longer look for source documents in paper form and increasingly had to perform tests and gather evidence electronically, thus their audit techniques had to undergo some changes (Bierstaker et al., 2001; Helms and Mancino, 1998).

The trend which caused the disappearance of the traditional audit trail continued and online systems and the Internet created an easier, cheaper way for data to be exchanged between systems. The *html* documents proved to be inadequate as it was difficult to extract and compare data because *html* only describes how data should be presented (Alles et al., 2004). Therefore, a language which could be intelligently manipulated and which would form a standard for data transfer was required. XML (eXtensible Markup Language) adds information about the document's content to its tags, thus it is easy to search, especially if digital agents are used. A subset of XML, XBRL (eXtensible Business Reporting Language) was created to describe business reporting information.

XBRL is useful for preparing, publishing, exchanging, acquiring and analysing accounting and business data, and provides a standardised method for transferring financial reporting information between different software applications (Alles et al., 2004; Srinivas, 2004). XBRL assigns tags to financial information, which allows computers to ''understand'' the data, while it can still produce human-readable reports. These tags are standardised by rules known as taxonomies. Taxonomies can contain rules (and tags) specific to certain industries or businesses as well as the Generally Accepted Accounting Principles (GAAP) rules. These may be region or country specific though (Pinkster, 2003).

XBRL enables continuous auditing by placing financial data in a format which is not proprietary to any specific software application, allowing any future continuous auditing system access to data on any software platform, running any software (which uses XBRL), in any country.

The concept of continuous auditing has been established in this section, this has explained *what* continuous auditing is. In the next section, *why* continuous auditing is necessary will be explained.

3. Motivation for CA technologies

In real-time accounting systems it has become desirable to have continuous assurances as to the condition of the information's integrity (Flowerday and von Solms, 2005). Furthermore, continuous assurances allow for corrective action to be taken sooner when a problem is found rather than in current auditing scenarios. To quote, ''*The focus of the audit will shift from manual detection to technology-based prevention*'' (Bierstaker et al., 2001).

Auditors aim to provide management with an opinion on subject matter for which management is responsible. In order to do this he/she will have to *validate* the *accuracy* of financial records and the *reliability* of the systems which store, transport and process those transactions. Looking at accuracy entails checking for fraud and error in transactions. There are well established auditing technologies which can assist in looking for *material* misstatements in financial records. Using these technologies within a continuous auditing system can extend their current effectiveness, as all transactions are analysed in real-time.

When assessing the reliability of the reports produced by the system, the auditor will look at confidentiality, integrity and availability and how this is ensured by the system of internal controls. Technology can be used to assess the internal control systems and see whether they are in line with prescribed norms. In the next section the nature of these technologies will be explored, explaining *how* the aims of continuous auditing can be met.

4. Technology aided tools and techniques

In order to verify that a real-time accounting system is producing reliable and accurate financial information, testing of controls must be done simultaneously with substantive tests of transactions (Helms and Mancino, 1999; Rezaee et al., 2001).

There are various tools and techniques which can aid in the analysis of transactions and internal controls. The tools are required to perform a variety of tasks. They can either be purchased software packages or auditor-designed routines (Rezaee et al., 2001). Collectively these tools and techniques are often referred to as Computer Aided Tools and Techniques or CATTs. An alternative acronym is CAATTs or Computer Aided Audit Tools and Techniques. CATTs have been used by auditors for many years and incorporate a wide variety of technologies, some of which are applicable to continuous auditing. In certain literature these have become known as Continuous Auditing Tools and Techniques. In this paper, ''CAATTs'' will refer to all of these collectively.

4.1. *Tools and techniques for analysing transactions*

To meet the requirements of an audit it is necessary to verify the accuracy of transactions to reveal fraud or error. Substantive tests of transactions must be performed. These will aim to obtain evidence showing possible material misstatements in the financial statements (South African Institute of Chartered Accountants, 2003). Two types of substantive tests are performed.

4.1.1. *Analytical procedures*

Analytical procedures involve performing comparisons of financial data to establish a relationship, often involving the

calculation of ratios. Analytical procedures not only indicate the possible existence of financial misstatements, they can also reveal how the client's industry and business function. When performed in the final phase of an audit, analytical procedures allow the auditor to comment on the reasonableness of transactions and the ability of the client to continue as a going concern.

CAATTs make analytical procedures more feasible and affordable than before (Rezaee et al., 2001). Many types of analytical procedures are too complex or time-consuming to be done manually. Using CAATTs also means that it has become possible to use larger sets of data when performing analytical procedures.

4.1.2. *Tests of transactions and balances*

The testing of transactions is often performed at the same time as testing controls. Transactions are tested continuously, throughout the financial year. This is done to discover whether material misstatements have occurred. In other words, to see whether erroneous or irregular processing of the transactions has taken place.

Testing transactions continuously throughout the year can help to reduce the number and/or complexity of tests of balances which need to be performed after balance sheet date (Rezaee et al., 2001). Tests done on balances are usually to collect evidence, on which the auditor can ground his or her opinion on fair representation of financial statements (Rezaee et al., 2002). When performing substantive tests of balances, Generalized Audit Software (GAS) tools are often used (Rezaee et al., 2001).

4.2. *Tools used in testing of internal controls and assessing risk*

In order to plan an audit, the auditor needs to be aware of the areas which carry the greatest risk and thus need the most scrutiny. This requires the auditor to look at the adequacy and effectiveness of internal controls within the system. According to the statement on auditing standards No. 80 (AICPA, 1996) CAATTs can be used for this purpose.

Testing of controls should also be ongoing. This allows the auditor to express an opinion as to how reliable the internal control system is. Knowing the reliability of the internal control system is important during the planning phase of an audit. The nature, timing and extent of substantive tests will be decided on accordingly (Rezaee et al., 2001).

In this section, the tools and techniques used for both analysing transactions and testing internal controls were elaborated, in the next section three of the models which use these tools and techniques will be explained.

5. Continuous auditing models

There are several suggested continuous auditing models, most are merely conceptual. Few seem to have been implemented in real-time systems. One of the early models was the Continuous Process Auditing System (CPAS) which was developed at AT&T Bell Laboratories. This is a methodology for internal auditing of large ''*paperless*'' real-time systems (Varsarhelyi and Halper, 1991). This model appears to have formed a basis for the later models.

In this paper, three of the better known models have been chosen for discussion. These all take somewhat different approaches and make use of different technologies.

5.1. *Continuous auditing building automated auditing capability (Rezaee et al., 2002)*

This is a conceptual framework for a continuous auditing system. It would be capable of running on a distributed client/server network and is also web-enabled for transmitting data to audit workstations. The model involves several steps.

Firstly, data are collected from transactional systems. This is done by linking to tables, via File Transfer Protocol (FTP), storage drives or via modem. The data are then stored on an audit server.

Once on the audit server, data are extracted from a variety of platforms and systems. Data standardisation is therefore required. Standards and formats are developed for storing data in the data warehouse/mart. The data are then transformed by cleaning, validating, restructuring the data and ''scrubbing'' with business rules.

An enterprise-wide data warehouse is not always needed, as it may be too expensive and complex. Instead, the required data could automatically be fed into several data marts. Data marts contain metadata which details the source transactions and the ETL (Extract Transform Load) process as well as the tests which take place. The metadata may for example include: detailed file definitions, business rules and transaction process flows.

Lastly, standardised tests are created to run within the data mart. The tests are created either to run continuously or at predetermined intervals. The tests are designed to automatically gather evidence and issue exception reports.

5.2. *Towards a paradigm for continuous auditing (Onions, 2003)*

To monitor the integrity of the data, Onions suggests *keystroke level* data examination. This basically involves monitoring database utilities and applications for commands which could cause fraud or error. This model addresses the testing of transactions in two ways.

Firstly, each transaction is audited and reported on as an isolated entity. This is done ''*ephemerally*'' – the transactions are tested at the time of entry. This is referred to as *transaction level* data examination. It ascertains whether each transaction fits the pre-specified rules for that transaction. These may be business rules or even rules dictating what actions are permissible for certain users. This is done in conjunction with performing certain analytical functions. Computer Assisted Audit Tools could be used. However, these operations would need to be performed on the transactions in real-time rather than batch. After the transaction has been examined it may be added to a data mine for possible further examination.

Secondly, the transactions are examined as a whole over a longer time period (perhaps even years). This examination looks for patterns in the transactions which could together result in fraud. This is known as the *transaction pattern level*

of data examination. Expert systems and rules based criteria would be employed. Rules would be similar to virus definitions and would be available for different industry types.

The problem when attempting to use expert systems is that each software package available has a different data schema. It would be very costly and time-consuming to create expert systems for each application. The solution would be to create a generic master file and transaction layout which could be used regardless of application data schema. This newly defined generic schema for a transaction would allow one expert system to trawl through the data mine. This schema would be defined using eXtensible Continuous Auditing Language (XCAL) which, similar to XBRL, is XML-based.

The model consists of four levels:

1. Transactions and data from various sources are entered for processing.
2. Transactions and keystrokes are mapped to XCAL schemas. This is done in real-time and is captured forensically on a daily basis.
3. Real-time CAATT processing is used to check transactions and keystrokes. Alerts may be sent to an Online Systems Audit Centre (OLSAC). Transactions are stored at this level for a day (but passed to level 4 where they are stored for years).
4. Expert systems look for patterns in the data.

5.3. *Continuous audit: model development and implementation within a debt covenant domain (Woodroof and Searcy, 2001)*

Woodroof and Searcy's model presents a conceptual model of continuous auditing. This model is limited in scope, as it is discussed in relation to debt covenant compliance. The model makes use of web-enabled technologies. It draws attention to the need for a reliable and secure system. The need for the production of *evergreen reports* is also discussed. Evergreen reports are reports which are generated on demand, usually viewed through a web site. The model is based on a database of transactions (journals and ledgers) on the client's system, with a web interface (on the auditors system) for the auditor to use.

This model is implemented in five stages:

1. A request is made for a report.
2. Agents and sensors within the client's system monitor the transaction data for exceptions to pre-specified rules. These exceptions are compared to the auditor-defined rules. This may trigger alarms, and alerts are sent to the auditor. The rules check the reliability of the system (possibly using continuous SYSTRUST), the fairness of the representation of financial reports and compliance to 3rd party contracts (like debt covenant agreements).
3. A digital agent on the auditor's system requests a digital agent on the client's system to retrieve the client's real-time balances of accounts via stored procedures in the client database.
4. If more information is returned than is needed, the digital agent extracts the information relevant to the ''contract'' (in this case debt covenant compliance). The information is checked for compliance, the actual event or process is checked against an acceptable standard for that event or process. If anomalies occur, these are flagged and the auditor is notified so that he/she may take action.
5. An evergreen report is generated and displayed to the loan officer. This details three levels of assurance. Level 1 is an assurance of reliability. If there is Level 1 exception, no further analysis is performed. Level 2 offers an opinion on the fairness of real-time financial statements. Level 3 provides an analysis of technical violations of 3rd party contracts (in this case debt covenant compliance is assessed).

Due to the reports being produced on demand (pull) (as opposed to being pushed to the user) this model is less suited to using XBRL-based reporting.

5.4. *Problems limiting use of CA systems*

One of the problems affecting continuous auditing solutions in real-time accounting systems is the varied data formats used. The ability to access and retrieve data from a variety of record sources, including legacy systems, is crucial to the creation of a continuous auditing system. This means that data will be in a variety of formats, with different file types and record systems. It becomes necessary to standardise these data. Unfortunately, this can be a complex and expensive process. Even more problematic is the risk of introducing errors such as duplicate records.

Technologies such as XBRL go a long way in creating a standard reporting format (Srinivas, 2004). Add to this, intelligent technology such as FRAANK (Financial Reporting and Auditing Agent with Net Knowledge) which can convert older reports into XBRL. This can create a way to compare non-XBRL data produced by legacy systems with newer XBRL reports (Kogan et al., 1998).

Until XBRL becomes widely implemented, using data marts to collect and assimilate data is an option. Onions (2003) also suggests adding XCAL, which would create a generic master file layout.

6. Evaluation of models

To evaluate these models one needs to consider at how accuracy and reliability are validated. In this context, *Accuracy* refers to how fraud and error in transactions are detected and how possible material misstatements in financial records are detected. *Reliability* is how confidentiality, integrity and availability of internal controls are examined. Further, these models will also be compared on *Real-time Processing* the *Reporting Method* used and the *Proposed Data Format* (Table 1).

It is apparent that the approaches of the three models differ slightly from each other, however, they all aim to function as close to real-time as possible. Some of the models use different technologies to achieve the same goal. For instance, detecting fraud and error may be accomplished by CAATS, digital agents or expert systems.

In the following section, suggestions will be made on how to draw together the tools and technologies used within these models to create comprehensive future CA systems.

Table 1 – Comparison of three continuous auditing models

	Rezaee et al.	Onions	Woodroof and Searcy
Accuracy (fraud and error) within transactions	Standardised audit tests are built into audit data marts. They run either continuously or at predetermined times. These gather evidence and then generate the relevant reports.	Transactions are checked both at time of entry and later. CAATTS (real-time, not batch). Expert systems (not in ''real-time'' but running continually).	Rule-based detection by digital agents. Data are analysed by devices integrated into the system.
Reliability of internal control system	CAATS are used. These include Integrated Test Facilities (ITFs) and parallel simulation. ITFs are used to verify correctness and completeness of processing. Parallel simulation tests assess effectiveness of control activities.	Parsing of keystrokes to detect database management utilities. Password control. Operating system's security. Audit logs. Web services verify information (e.g. new supplier's credit history checked).	Adapt and apply SYSTRUST principles. Web-based valuation sites. Must be in the auditor-defined rules for the digital agents.
Real-time	Real-time processing is the aim for this system.	All proposed systems run in parallel with operational systems in real-time.	Real-time reporting is one of the aims of this model. To this end, information must be collected and monitored in real-time.
Reporting method	Web-enabled data delivery of data to auditors' workstations, where reports can be generated (possibly by Generalized Audit Software).	Graded alerts sent through Virtual private networks (VPNs) to audit department/OLSAC. The alerts are graded by gravity (three levels).	Three levels of reporting, alerts are sent to the auditor via email. Level 1: reliability of the system or security of the transmission. Level 2: transactions and processes. Level 3: technical violation of 3rd party agreement. 3rd party and auditor notified by email. Evergreen reports are produced on demand through a web interface – information *pull* approach (as opposed to XBRL reporting this is push reporting method).
Proposed data format	Data mart Data warehouse XBRL	XCAL Data marts	Does not interface with legacy systems.

7. The future of CA technology

To meet the requirements of continuous auditing, any comprehensive CA model would need to address both *internal control testing* and *testing of transactions*. A dual-pronged approach, where both the system (internal controls) and data (transactions) are tested, simultaneously would be the ideal.

7.1. *Internal control issues*

There is a need to collect evidence on the quality and integrity of an electronic system in producing reliable and accurate financial information. Technology must aid in verifying the integrity of data, because the conclusions in auditors' reports must be based on accurate and reliable data in order to be deemed trustworthy (Wessmiller, 2002).

There is also a need to ensure the security of the system. A system which is not secure is not reliable. Ensuring security would involve examining internal controls. If the system is not reliable, the results from that system may not be viewed as trustworthy. Woodroof and Searcy (2001) suggest SYSTRUST (or a CA derivative of SYSTRUST). COBIT Guidelines in conjunction with ISO 17799 could also be used. COBIT could address internal control related issues (The IT Governance Institute, 2005). ISO 17799 would aid in addressing information security issues (ISO/IEC 17799, 2005). Thus, a number of best practices and/or standards exist to address the security issues.

Rezaee et al. (2002) mention Concurrent Audit Techniques for testing effectiveness of a client's internal controls. Concurrent Audit Techniques include SCARF (Systems Control and Review Facility) and the snapshot approach, where SCARF functions as an exception reporting system. It captures transactions meeting certain criteria (defined by the auditor) by using Embedded Audit Modules. The captured transactions are set aside for later review by an auditor. Embedded Audit Modules and the SCARF approach could be used to create alerts regarding the status of internal control systems by checking if controls are implemented.

7.2. *Transactions*

Processing of transactions should occur in several stages. Various technologies help throughout these stages. Firstly, transactions from a variety of sources are extracted. Not all records or fields may be required, digital agents and stored database procedures could be used to pull out only the necessary data. The creation of data marts and data warehouses may be desirable, a capable Database Management System (DBMS) would be required.

Once data have been collected and transformed, the transactions need to be assessed. Transactions are individually assessed for integrity (errors and fraud) and validity (business rules). CAATTS can be made to run as close to real-time as possible, while the data are flowing through the application system. Both analytical procedures and substantive testing should be applied to look for fraud and error. Common IT-based fraud schemes often involve the billing system, payroll system and check tampering. These schemes often need to be identified at the transaction data level, as they can depend on groups of transactions within the system. Examples include ghost vendors, ghost employees and exploiting voids and returns (Taylor, 2005).

Often a Database Management System (DBMS) forms an important part of a system. An example of a commonly used DBMS is Oracle. Oracle is a DBMS which allows ''triggers'' to perform tasks when certain criteria are met. Triggers are useful for creating logs for system events (Finnigan, 2003). Triggers in a database can be used in the same way as Data Query Modules (DQMs). DQMs are macros or programs built using audit software, they perform queries to answer a specific question posed by the auditor. For example, DQMs could be used to look for fraudulent travel allowance claims of employees. Employee swipe card data could be compared to dates on tour and travel reports. If the employee's card was swiped, and they were at the office, they are most likely claiming travel allowance for extra days (Dalal, 2000). If the entry of a new travel claim triggered the execution of the relevant DQM, an instant audit would occur.

At this stage, alerts could be produced. For example control agents may be used to alert auditors if transaction values change too much from the norm. The relevant transaction data need to be collected for future forensic analysis – possibly by moving the data to a secured partition or dedicated audit server. This may be achieved using FTP and tape or other large-capacity storage devices. Digital agents then examine transactions and select those that should be set aside for later analysis. CIS (Continuous and Intermittent Simulation) could also be used for deciding which transactions require more examination.

The data may then need to be standardised. This may be done either on the audit server or in the source application, depending on the cost of processing. For example, the cost of processing on mainframes is more expensive. The data can then be aggregated in data marts or data mines. Data mines are expensive and normally only larger organisations can afford them. Data marts are often used by smaller organisations, or in larger organisations for one specific focus area, for example, Human Resources, Accounting data, etc. (Rezaee et al., 2001). Smaller organisations could also make use of XCAL, as suggested by Onions (2003).

Stored data may then be checked for groups of transactions and the cumulative effects of a series of transactions. Expert systems and digital agents may be very useful for this purpose. Analytical procedures could also be used to create ''norms'' which can be used as a benchmark. The results of these tests are then used to create reports and alerts, which are sent to auditors by encrypted email. The alerts may also be sent via VPNs. A grading system for alerts, showing the possible impact of the anomaly, may be important to the auditors.

It may also be necessary to be in contact with an outside auditing bureau for verification and validation. For example the Online Systems Audit Centre (Onions, 2003). The Online Systems Audit Centre is a group of auditing professionals, which monitor and investigate alerts online. VPNs can be used for this purpose. Web-enabled agents like FRAANK, and Web Services may also provide useful, secure ways to communicate (Kogan et al., 1998; Murthy and Groomer, 2004).

8. Conclusion

Within real-time accounting systems, real-time assurances are not only desirable, but are possible. Technologies which enable the provision of real-time assurances are becoming commonplace. This includes technologies which test internal controls and those related to testing transactions. Many of these technologies are not new, for example CAATTS, including GAS, EAMs and ITFs, which are being applied in new ways to achieve continuous auditing. Some innovative technologies, such as AI technologies allow every transaction to be inspected, instead of just a sample. These available technologies need to be brought together in a way which makes full use of each of them.

Models have been suggested for this purpose, however, most are only conceptual. These models can be adapted and

adjusted in order to provide the auditor with the reliable and accurate results he or she desires. A possible reason for the lack of comprehensive continuous audit models may be the result of the problems related to the variety of data formats which exist and the availability of audit data. Technologies such as XBRL contribute to solving the problem, but further enhancements are definitely envisaged.

A comparison of the three most prominent continuous auditing models is tabulated. The models are compared according to a set of criteria. These include: how accuracy and reliability are evaluated, what the reporting method is, and how close to real-time the model functions, and the proposed data format. These findings highlight the core aspects of the three models and can be used as a foundation on which to build future continuous auditing solutions.

REFERENCES

Alles M, Kogan A, Vasarhelyi M. Real time reporting and assurance: has its time come? Available from: http://raw.rutgers.edu/continuousauditing/Real_Time_Reporting_-_ICFAI1.doc; 2004 [retrieved 12.05.2005]

American Institute of Certified Public Accountants (AICPA). Amendment to statement on auditing standards no. 31, evidential matter: SAS 80; 1996.

Bierstaker JL, Burnaby P, Thibodeau J. The impact of information technology on the audit process: an assessment of the state of the art and implications for the future. Managerial Auditing Journal 2001;16(3):159–64.

CICA/AICPA. Continuous auditing: research report. Canadian Institute of Chartered Accountants; 1999.

Dalal C. Advanced use of audit software in audit and fraud detection – audit software: an indispensible tool in the new millennium. Ineternal Auditors 2000;3(February 1).

Finnigan P. Introduction to simple oracle auditing. Available from: http://www.securityfocus.com/print/infocus/1689; 2003 [retrieved 10.04.2006].

Flowerday S, von Solms R. Real-time information integrity = system integrity + data integrity + continuous assurances. Computers and Security 2005;24:604–13.

Helms GL, Mancino J. Wave good-bye to the paper trail. Electronic auditor. Available from: http://www.aicpa.org/pubs/jofa/apr98/helms.htm; 1998 [retrieved 7.03.2005].

Helms GL, Mancino JM. The CPA & the computer: information technology issues for the attest, audit, and assurance services functions. Available from: http://www.nysscpa.org/cpajournal/1999/0599/departments/cpac.html; 1999 [retrieved 3.05.2005].

ISO/IEC 17799. Information technology – security techniques – code of practice for information security management. International Organization for Standards. Available from: http://www.iso.org/iso/en/ISOOnline.frontpage; 2005.

Kogan A, Nelson K, Srivastava R, Vasarhelyi M, Bovee M. Design and applications of an intelligent financial reporting and auditing agent with net knowledge. Available from: https://kuscholarworks.ku.edu/dspace/bitstream/1808/141/1/srivastava.pdf; 1998 [retrieved 10.05.2005].

Murthy US, Groomer SM. A continuous auditing web services model for xml-based accounting systems. International Journal of Accounting Information Systems 2004;5:139–63.

Onions RL. Towards a paradigm for continuous auditing. Available from: http://www.auditsoftware.net/community/how/run/tools/Towards%20a%20Paradigm%20for%20continuous%20Auditin1.doc; 2003 [retrieved 1.04.2005].

Pinkster R. XBRL awareness in auditing: a sleeping giant? Managerial Auditing Journal 2003;18(9):732–6.

Rezaee Z, Elam R, Sharbatoghlie A. Continuous auditing: the audit of the future. Managerial Auditing Journal 2001;13(3):150–8.

Rezaee Z, Sharbatoghlie A, Elam R, McMickle P. Continuous auditing: building automated auditing capacity. Auditing: A Journal of Practice and Theory 2002;21(1):147–63.

South African Institute of Chartered Accountants. SAICA handbook – auditing. 2003/2004 ed., vol. 2; 2003.

Srinivas S. Road map to XBRL adoption as a new reporting model. Information Systems Control Journal 2004;1.

Taylor P. The perils of systems-based fraud. IT Audit 2005;8(January 15).

The IT Governance Institute. (COBIT) Control objectives for information and related technology. 4th ed. USA: The IT Governance Institute; 2005.

Vasarhelyi MA. Concepts in continuous assurance. Available from: http://raw.rutgers.edu/continuousauditing/conceptsincontinuousassurance13final.doc; 2002 [retrieved March, 2005].

Varsarhelyi MA, Halper FB. The continuous audit of online systems. Auditing: A Journal of Practice and Theory 1991;10(1).

Wessmiller R. Facing the data integrity challenge. Available from: http://www.theiia.org/itaudit/index.cfm?fuseaction=print&;fid=440; 2002 [retrieved 20.04.2005].

Woodroof J, Searcy D. Continuous audit: model development and implementation within a debt covenant compliance domain. International Journal of Accounting Information Systems 2001;2:169–91.

Stephen Flowerday is currently a final year full-time doctoral student at the Nelson Mandela Metropolitan University in South Africa. His research focus is on providing real-time assurances for information integrity. This is within the domain of corporate governance and information security management. In addition to his studies he lectures part-time and before entering the academic field he had a successful career in management consulting.

Adrian Blundell is presently completing a full-time Master's degree in Information Technology at the Nelson Mandela Metropolitan University in Port Elizabeth, South Africa. He is researching the roles of various technologies within continuous auditing systems. His qualifications include a National Diploma IT from the Port Elizabeth Technikon and a B. Tech IT from Nelson Mandela Metropolitan University.

Professor Rossouw von Solms is the Director of the Institute for ICT Advancement at the Nelson Mandela Metropolitan University in South Africa. He holds a PhD from the Johannesburg University. He has been a member of the International Federation for Information Processing (IFIP) TC 11 committee since 1995. He is a founder member of the Technikon Computer Lecturer's Association (TECLA) and is an executive member ever since. He is also a vice-president of the South African Institute for Computer Science and Information Technology (SAICSIT). He has published extensively in international journals and presented numerous papers at national and international conferences in the field of Information Security Management.