Audit Automation for Implementing Continuous Auditing: Principles and Problems

Michael G. Alles
Alexander Kogan
Miklos A. Vasarhelyi

Rutgers Business School

Department of Accounting,
Business Ethics & Information Systems
180 University Ave
Newark, NJ 07102

Version: August 20, 2008[*]

Abstract: When implementing continuous auditing, experience indicates that auditors will likely attempt to first automate the processes that they already use, are comfortable with and are already accepted for external auditing and reporting purposes rather than trying to start from scratch, especially when dealing with audits of ongoing operations. However, because of the experience of low productivity and failed expectations with prior technology implementations that gave rise to the argument for business process reengineering in Hammer's (1990) article, "Don't Automate, Obliterate" special care needs to be taken when change is brought about by automation. As we argue in this paper, not only must audit automation be undertaken systematically, it also has to incorporate reengineering in the more limited sense of first transforming manual audit processes to facilitate their automation. This is not full blown reengineering of the clean sheet sort, but this hybrid approach is one that is more manageable—and marketable—from a change management perspective, and more likely to lead to a positive outcome. The key for avoiding the potential downsides of automation, though, is to have a clear understanding of what audit automation is trying to achieve and follow a methodical procedure to achieve those goals.

Keywords: audit automation, continuous auditing, internal audit, audit systems.

[*] Comments welcome. Please address them to kogan@rbsmail.rutgers.edu.

1. Introduction

Alles et al (2006) described a feasibility study undertaken by the IT Internal Audit department at Siemens Corporation, working with the authors, to create a continuous auditing (CA) system by automating the largely manual audit of its SAP systems. Since that initial study, the field of continuous auditing has rapidly developed, with vendors offering sophisticated IT products that facilitate CA implementation, as well as many other firms having begun to develop homegrown CA processes. This paper builds on the authors experience with such vendors and firms, as well work we undertook with Siemens on its second generation CA implementation. While the details of that particular case study is described elsewhere (Teeter and Brennan, 2008), in this paper we step back to draw general conclusions about the challenges that auditors will face when automating existing audit procedures for a CA environment, as well as the opportunities that they now have with new CA-enabling technologies.

As with Alles et al. (2006), our focus is on automation of an existing, predominantly manual audit process. While in years to come new CA audits may be created with a blank sheet approach, the experience of much technology implementation from mini-computers to ERP suggests that the change process is likely to be incremental rather than disruptive. Hence, as at Siemens, auditors will likely attempt to first automate the processes that they already use, are comfortable with and are already accepted for external auditing and reporting purposes rather than trying to start from scratch, especially when dealing with audits of ongoing operations. Moreover, audit standards have been largely written for a world in which technology may be an enabler, rather than the driver of audit processes as it is in CA, which again implies that the current need is for an understanding of automation as the primary mechanism used to bring about CA.

At the same time, precisely because of the experience of low productivity and failed expectations with prior technology implementations that gave rise to the argument for business process reengineering in Hammer's (1990) article, "Don't Automate, Obliterate" special care needs to be taken when change is brought about by automation. As we argue in this paper, not only must audit automation be undertaken systematically, it also has to incorporate reengineering in the more limited sense of first transforming manual audit processes to facilitate their automation. This is not full blown reengineering of the "throw

away the [manual] rule book" sort, but this hybrid approach is one that is more manageable—and marketable—from a change management perspective, and more likely to lead to a positive outcome. The key for avoiding the potential downsides of automation, though, is to have a clear understanding of what audit automation is trying to achieve and follow a methodical procedure to achieve those goals.

We begin by considering the case for audit automation and its relation to continuous auditing.

2. Drivers and Objectives of Audit Automation

Half a century has passed since the original utilization of computers in business. Even in those early days the tremendous potential impact of automation on accounting and auditing was understood and brought to the attention of the profession and academia (Keenoy, 1958). What was in the beginning a limited scope deployment focused on carefully selected business areas (such as payroll processing and inventory control) and utilized primarily by large enterprises, has since become ubiquitous for most business entities, and is now inseparable from doing business. Most business processes today are automated to various degrees, and businesses continue to invest in maintaining and expanding this automation through the acquisition of computer and telecommunication technologies and various enterprise systems, such as enterprise resource planning, data warehousing, supply chain management, and customer relationship management.

Automation of business processes has inevitably led to changes in auditing procedures and standards. Starting with the original release in 1973 of SAS 3 ("The Effects of EDP on the Auditor's Study and Evaluation of Internal Controls"), the utilization of modern information technology has made its way into the audit process, prompted by growing availability and decreasing cost of personal computing and office automation software (word processors and spreadsheets), as well as generic statistical software and dedicated computer-assisted audit techniques, such as Audit Command Language. These early deployments demonstrated the potential of IT in making labor-intensive repetitive audit work more efficient. The potential impact of audit automation on the changes in the audit process was originally researched almost quarter century ago by Vasarhelyi (1984). Recent empirical studies (Banker et al., 2002) have shown that IT-enabled audit automation indeed leads to significant productivity gains.

automation involved primarily CCM, the general principles and problems of audit automation generalized from that experience apply to both aspects of CA.

3. Audit Automation Change Management

The audit profession is inherently conservative given that its entire value added comes from the auditor's credible claims of objectivity and reliability. As a consequence, auditing processes, even more so than other business processes, have a tremendous amount of inertia. It follows that any audit automation project, as with any major change initiative in such circumstances, will have numerous barriers to change to overcome. This is why it is critical to ground audit automation projects in sound business process change methodologies developed in the management literature as a result of extensive experience with large scale IT implementations (see e.g., Davenport and Short, 1990; Wastell et al., 1994; Kettinger et al., 1997; Reijersa and Liman Mansar, 2005).

As research on ERP implementation success factors has convincingly shown (Umble et al., 2003; Botta-Genoulaz et al., 2005), for an automation project to even get launched, let alone succeed, senior executive champions have to take ownership of the project, both at the internal audit level, and at their reporting level in the C-suite or the audit committee. In the case of Siemens the champion that brought the authors on board was the then head of the IT Internal Audit Department. The fact that we are increasingly coming across executives at firms with titles such as "Associate Director, Continuous Assurance" (in the case of BD Corporation) indicates that such champions are becoming institutionalized in firms as CA goes mainstream.

The first critical task of audit automation champions will be to identify and engage project stakeholders. In addition to internal auditors, these stakeholders will include business process owners and IT personnel. Again, the use of such multifunctional teams is a standard recommendation of change management theory, but in the case of audit automation the problem is compounded by the need of internal audit to be aware of the needs of the external auditor, while also balancing the demands of the IT process owners and line managers. The composition of audit automation teams must reflect the multi-faceted nature of the task at hand.

The reason for having a high powered team with a senior level champion is obvious when considering the complexity inherent in automating audit processes initially designed to be done largely manually. In our experience, even very experienced auditors differ in how such procedures are carried out in practice, which translates into differences in how to transform the process into an automated one, what the objective of the process should be and how much weight should be placed on a particular process or on a possible compensating control.

A powerful way of increasing the quality and reliability of audit automation results would be to diversify risk and bring out differences by utilizing duplicate audit automation teams, and then comparing the resulting automated audit programs. Resolving the inevitable disagreements between the duplicate teams would greatly improve the final automated program. While this approach is often utilized in academic research (e.g., to make sure that human responses are coded properly), we have yet to come across any instance in which such a procedure has been adopted. It is simply too expensive, both in terms of human resources and time to be feasible for the vast majority of enterprises. Therefore, alternative measures have to be utilized to assure the quality of the automated audit program both during the automation process and to verify the completed product.

One such alternative is for the automated audit procedure developed by the automation team to be verified independently by experienced auditors who took no part in developing it. A vitally important check on the audit automation process is the need to satisfy the external auditor, and to retain their reliance on the internal audit process. As, in accordance with SAS 65 ("The Auditor's Consideration of the Internal Audit Function in an Audit of Financial Statements"), the external auditor evaluated the quality and effectiveness of the original manual audit process, such evaluation can encompass the automation process, the finished automated audit program, or ideally, both. In either case, demonstrating that the automation team followed a systematic procedure is an essential element in satisfying the external auditor.

4. Formalizing and Reengineering the Audit Program

As Alles et al. (2006) indicated, before audit procedures can be automated, they must first be formalized: *"Automation requires formalization of audit procedures. Approved audit programs are not*

contracts. At the same time, the possibility of formalization is often underestimated, and when an earnest effort is made to formalize audit procedures, the results often exceed the most optimistic expectations. Alles et al. (2006) concluded this much in their study of Siemens first generation CA implementation, but importantly, Teeter and Brennan (2008) indicated that recent advances in CA-enabling software, such as Approva, have increased the scope for formalization and audit automation.

Siemens' internal audit methodology for SAP facilities involves the carrying out of several hundred of "audit action sheets" by internal auditors at the auditee site. Alles et al. (2006) indicated that about 25% of the audit actions could be fully automated due to their deterministic nature. But Teeter and Brennan (2008), in their study of the CA initiative based on an Approva system foundation concluded that about 68% of the actions could be automated to some extent. Considering that some of these automated steps would be performed in a daily monitoring mode (as opposed to the 18 to 24 month cycle of SAP audits) the strength of its evidence would be much stronger and conceivably could replace much of the residual 32% non-automated evidence. Such replacements have been shown in some past experience to be the main benefit of audit automation (Fischer, 1996).

5. Baseline Monitoring ("Baselining")

Baseline monitoring or "baselining" for short is a well established procedure in configuration management and IT security, defining the "what should be" state that is used as a benchmark against which the current state can be compared. As applied to enterprise systems, baseline is defined as a set of system configuration and business process settings at a given, reference point of time.

Baselining is facilitated in audit automation through the use of the "snapshot" feature of audit automation software (such as Approva), which does precisely what the name suggests (albeit only for the SAP tables that the program monitors). The use of baselining allows the automation of audit procedures that would be difficult to formalize explicitly, by allowing for a simple before/after comparison. The price to pay for this approach towards automation is the extensive manual effort to initially verify all the values in the baseline to make sure that they are appropriate. This verification relies on human judgment typically supported by manual techniques such as interviewing, investigations and observations. However,

Datar (2004) state: *"Interactive controls are particularly important in control theory because it is they that tell senior managers when they need to change strategy—and so, when they have to alter the belief, boundary and diagnostic controls that put that strategy into practice. In other words, interactive controls make the entire control architecture dynamic, enabling it to evolve as underlying conditions change, or as core assumptions are proved invalid."*

The application to baselining is to draw the analogy of the baseline to a boundary control and to develop an analytic engine for alarms as an interactive control that would indicate when exceptions are being caused by a change in the underlying operations, thus necessitating an evolution of the automated audit, as opposed to an anomaly caused by inappropriate behavior by the auditee. In other words, baselining and their technological enabler, the "snapshot" feature of monitoring software, are simply tools, means towards an end. Putting them to work as an automated audit procedure requires that they be overlaid with an effective control framework that will enable their transformation into a dynamic monitoring process. It is possible that combining Simons' framework with the very powerful snapshot capabilities of modern audit enabling software will lead to the development of a highly capable CA system that goes beyond first-generation audit automation.

However, it needs to be reiterated that the initial manual verification of baseline values is a critical stage of audit automation. Any mistake made at this stage will be leveraged since the system will automatically perpetuate the mistake indefinitely. This is an indication of the different set of risks that can arise in automated systems. While in manual auditing there is always a chance that a mistake made during a particular audit cycle can be corrected during subsequent periods, in baselining there is only one chance during the initial verification stage to get things right.

Another critically important issue is the security of baseline—both in its definition and its current values. If one can compromise the security of the baseline and manipulate the definition of the baseline, say by removing from it certain parameters or by changing certain values in the baseline, then one can potentially open gaping holes in the system without anybody ever noticing. Thus, securing the baseline must be a well-developed feature of an automated auditing system.

6. Architecture of Automated Auditing

The other side of the issues discussed above is the necessity to protect the EAM or mobile agent auditing code against possible manipulation by the enterprise platform. Given that the superuser privileges for the enterprise system are held by the enterprise IT personnel, the integrity of the audit code processing is always in question since it is the objective of this code to check on the enterprise system and its personnel. This problem has been discussed in the literature under the name of the *malicious host problem* (Jansen and Karygiannis, 1999; Claessens et al., 2003), and it is considered to be extremely difficult (if not infeasible). While there have been proposed numerous ways of dealing with it (see e.g., Stengel et al, 2005; Futoransky et al, 2006; Shao and Zhou, 2006; Topaloglu and Bayrak, 2008), and there are some quite complex ways of detecting the problem, no solution has been universally accepted as being able to prevent the host from interfering with code's execution.

The extreme difficulty (if not impossibility) of protecting the EAM or mobile agent auditing code from possible manipulation by the enterprise platform puts in question the integrity of results provided by this auditing code. This lack of trust in the audit results outweighs the benefits of the resident code described above, and serves as one of the critical reasons for basing automated auditing architecture on remote monitoring of enterprise systems.

7. Handling, Evaluation and Integration of Audit Evidence

While it is the ultimate objective of any business enterprise to have a totally reliable control system that will not have any exceptional events or anomalous situations, this ideal is never achieved, and the auditing system will be generating alarms caused by anomalies and exceptions. These alarms will be delivered by automatic means (e-mail, instant and wireless messaging) to the appropriate auditors and enterprise personnel responsible for resolving them. While the automated auditing system keeps track of each event, it is essential to have an automated closed loop process for capturing information about the corrective actions and assuring that these actions resolve the underlying problem.

As the results of resolving exceptional events and anomalous situations identify various control failures of the enterprise system, the auditing system should have a built-in mechanism for evaluating how significant these failures are, and making these evaluations available in a timely manner to the relevant stakeholders (auditors and upper management). To make such automatic evaluations possible, the procedures in the automated auditing

system have to be organized in accordance with the enterprise risk model to associate appropriate risks to various control failures.

While individual evaluations of control failures are no doubt important, large enterprises in particular would be interested in aggregating audit evidence to see the "big picture" of current enterprise exposure. The development of sound theoretical methodology for measuring internal control performance and aggregating audit evidence that would be practically applicable in modern large enterprises presents serious challenges. Theoretical studies of internal controls design and evaluation undertaken over the years (Cushing, 1974; Srinidhi, 1988; Vasarhelyi and Srinidhi, 1989; Knorr and Stormer, 2001) can provide a foundation for future practical developments which are yet to come. In the meantime, various ad hoc solutions and simplifying assumptions can be utilized to build a continuous auditing dashboard that in real time provides an aggregate view of enterprise control problems.

8. Software for Audit Automation

While it is certainly possible to design, develop and implement a custom-made automated auditing system in house, the expense and expertise requirements of such a project make it prohibitively expensive, if not outright infeasible, for the vast majority of cases. It is therefore not surprising that there is an emerging industry of packaged software developed to support audit automation or at least some of its aspects.

A convenient way of categorizing the current software offerings is in accordance with the breakdown of CA as consisting of CCM and CDA. While the vendors are attempting to integrate in their packages as many features as possible, they still typically exhibit strength in one of the two components. The well-established CAATs vendors ACL and CaseWare IDEA have extended their products to position them as continuous monitoring solutions. ACL in particular has invested significant efforts into providing what they call "continuous controls monitoring" solutions. Despite the name, in the terminology of this paper these solutions should be categorized as CDA since the substance of their tests is transaction verification and analysis focused on making inference about the functioning of controls (as opposed to direct tests of controls through monitoring of their settings). A relative

functionality that would bring these solutions closer to the fully developed CCM and/or CDA systems.

Automating a manual audit program requires a significant startup expense. This fixed cost may become a significant hurdle in the way of audit automation if an enterprise has no way of amortizing this cost over different enterprise units, since automation of highly specific audit procedures for different enterprise units can incur prohibitive costs. Automation will be scalable across the enterprise only if the repetitive audit procedure automation costs are eliminated.

There are a number of strategies for making audit automation scalable. The most immediate one is the parameterization of automated audit procedures. Given the expected significant homogeneity of enterprise business processes, it is likely that one can make automated audit procedures sufficiently generic by introducing various parameters describing systems, processes and business artifacts. Then the implementation of automated audit in additional enterprise units can be reduced to properly configuring the already developed system by assigning the appropriate parameter values.

In addition to parameterization, scalability of the audit program can be enhanced through hierarchical structuring of automated audit procedures – from the most generic audit procedures applicable across the enterprise to the more specific ones for major units and subunits. The feasibility of such structuring is enabled by the natural hierarchy of business enterprises and the risk-based top-down approach towards audit program development. This will also facilitate audit program maintenance through hierarchical updates: given a change in the processes at a certain node of the enterprise hierarchy, only the audit procedures in the hierarchical sub-tree rooted at this node will have to be reviewed and, possibly, revised.

9. Securing Continuous Auditing

Based on the analysis above, it is likely that an automated auditing system will be implemented as a MCL hosted on its own platform. To assure the integrity of its results, this system must be thoroughly secured. One of the issues that critically affect this system security is the control of the continuous auditing software, and its associated hardware,

which can be either under the authority of the auditee's IT department, or under the internal auditors themselves. Although the latter is intrinsically more secure, there are numerous practical matters that can favor the former. While we have come across instances in which the internal auditors have their own CA software (albeit, running on the firm's general IT systems), the cost and complexity of software such as Approva makes it far more likely to be entirely run by the firm's IT department (or outsourced to the vendors or third parties) with access provided to the auditors. In this case, one has to pay particular attention to access security.

Logical access security of the auditing system is even more critical since any compromise of the system can potentially be used to cover up for undesirable events in the enterprise system. As Alles et al. (2002, 2004) argued, automated audit systems are in one sense more vulnerable than manual systems due to the lack of "another-pair-of-eyes" controls and the leveraging of a mistake—intentional or otherwise—every time the CA system runs. Maintaining logical security is particularly problematic since it requires advanced system management IT skills which may not be easily available in internal auditing. The super-user privileges in the auditing system are figuratively speaking the "keys to the kingdom", and their safeguarding requires utmost attention, which is why this is a key control for SOX 404 certification.

To mitigate this exposure, comprehensive logging of all super-user activities should also be implemented and constantly monitored by the internal auditors, as Alles et al. (2004) proposed. An important control over the security of the auditing system consists in exporting its settings and cryptographic check-sums of its code to an external storage facility or/and non-volatile storage medium. Then these setting can be imported back into the CA system as needed, and the cryptographic check-sums of the system code can be recalculated to verify the integrity of the CA system. This is obviously a de-facto tertiary monitoring process, and the administration of this process should be done by dedicated internal audit personnel.

10. Concluding Remarks

The practice of audit automation will be strongly influenced by the ongoing software development and maturing of the field of GRC. AMR Research projects that spending on

quality of work the automated auditing systems are providing. It is likely that it will also result in significant changes in the nature of audit procedures performed by the external auditor, and these changes in turn may lead to structural changes in audit engagements, and may, with time, reshape the external audit as we know it today.

References

1. Alles, M.A., and Datar, S. 2004. Cooking the Books: A management-control perspective on financial accounting standard setting and the section 404 requirements of the Sarbanes-Oxley Act. *International Journal of Disclosure and Governance,* Vol. 1, No. 2, 119–137.
2. Alles, M.G., A. Kogan, M.A. Vasarhelyi. 2002. Feasibility and economics of continuous assurance. *Auditing: A Journal of Practice and Theory*, Vol. 21, No. 1 (March), 125–138.
3. Alles, M.G., A. Kogan, M.A. Vasarhelyi. 2004. Restoring Auditor Credibility: Tertiary Monitoring and Logging of Continuous Assurance Systems. *International Journal of Accounting Information Systems*, Vol. 5, No. 2 (June), 183–202.
4. Alles, M.G., G. Brennan, A. Kogan, M. A.Vasarhelyi. 2006. Continuous Monitoring of Business Process Controls: A Pilot Implementation of a Continuous Auditing System at Siemens. *International Journal of Accounting Information Systems*, Vol.7, 137–161.
5. Alles, M.G., A. Kogan, M.A. Vasarhelyi, J. Wu. 2008a. *Continuous Data Level Auditing Using Continuity Equations.* Unpublished working paper, Rutgers Business School.
6. Alles, M.G., A. Kogan, M.A. Vasarhelyi. 2008b. Putting continuous auditing theory into practice: Lessons from two pilot implementations. *Journal of Information Systems*, forthcoming.
7. Banker, R. D., H. Chang, and Y. Kao. 2002. Impact of information technology on public accounting firm productivity. *Journal of Information Systems* 16 (2): 209–222.
8. Bedard, J. C., D. R. Deis, M. B. Curtis, J. G. Jenkins. 2008. Risk monitoring and control in audit firms: A research synthesis. *Auditing: A Journal of Practice & Theory*, Vol. 27, No. 1 (May), 187–218.
9. Bell, T.B., F. Marrs, I.Solomon, H. Thomas. 1997. *Auditing Organizations Through a Strategic-Systems Lens: The KPMG Business Measurement Process.* KPMG, Montvale, NJ.
10. Botta-Genoulaz, V., P.-A. Millet, B. Grabot. 2005. A survey on the recent research literature on ERP systems. *Computers in Industry*, Vol. 56, No. 6 (Aug.), 510–522.
11. Claessens, J., B. Preneel, J. Vandewalle. 2003. (How) can mobile agents do secure electronic transactions on untrusted hosts? A survey of the security issues and the current solutions. *ACM Transactions on Internet Technology*, Vol. 3, No. 1 (February), 28–48.
12. Cooke, N. J. 1994. Varieties of knowledge elicitation techniques. International *Journal of Human-Computer Studies*, Vol. 41, No. 6 (December), 801–849.