CONTINUOUS MONITORING OF BUSINESS PROCESS CONTROLS: A PILOT IMPLEMENTATION OF A CONTINUOUS AUDITING SYSTEM AT SIEMENS

Michael G. Alles
Associate Professor
Rutgers University

Gerard Brennan, PhD
Director of IT Audit
Siemens Corporation

Alexander Kogan
Professor
Rutgers University

Miklos A. Vasarhelyi
KPMG Professor of AIS
Rutgers University

Version: September 1, 2005[*]

[*] Rutgers Business School, 180 University Avenue, Newark NJ 07102. Comments are welcome and may be addressed to Alex Kogan at kogan@andromeda.rutgers.edu. We thank the KPMG CAR-Lab at Rutgers Business School and Siemens Corporation for support. All views expressed in this paper are those of the authors alone.

CONTINUOUS MONITORING OF BUSINESS PROCESS CONTROLS: A PILOT IMPLEMENTATION OF A CONTINUOUS AUDITING SYSTEM AT SIEMENS

In this paper we report on the approach we have developed and the lessons we have learned in an implementation of the monitoring and control layer for *continuous monitoring of business process controls* (**CMBPC**) in the US internal IT audit department of Siemens Corporation. The architecture developed by us implements a completely independent CMBPC system running on top of Siemens' own enterprise information system which has read-only interaction with the application tier of the enterprise system. Among our key conclusions is that "formalizability" of audit procedures and audit judgment is grossly underestimated. Additionally, while cost savings and expedience force the implementation to closely follow the existing and approved internal audit program, a certain level of reengineering of audit processes is inevitable due to the necessity to separate formalizable and non-formalizable parts of the program. Our study identifies the management of audit alarms and the prevention of the alarm floods as critical tasks in the CMBPC implementation process. We develop an approach to solving these problems utilizing the hierarchical structure of alarms and the role-based approach to assigning alarm destinations. We also discuss the content of the audit trail of CMBPC.

Keywords: Continuous auditing, continuous monitoring of business processes, controls, control settings, monitoring, formalization, automation, reengineering.

1. Introduction

> The experience with the evolution of new technologies and business processes suggest that CA will initially be used to do no more than automate existing audit procedures, and thereby take full advantage of the capabilities that it has in the new ERP based environment… [The] second stage of its evolution [will be reached] when audit processes are reengineered to exploit the underlying technological capabilities to the fullest…
>
> However, to reach that stage will require more than technology implementation. For one thing, it will necessitate auditors actually examining their processes to see if they are susceptible to process mapping and reengineering… At the same time, continuous analytic monitoring will intrude into the internal control arena, especially since it is built on the firm's own ERP systems…
>
> While the theoretical work in CA has made progress, the field has been hindered by the lack of a proper set of experimental and empirical research.
>
> From Vasarhelyi et al (2004), pp. 19-20.

Providing assurance in the modern business environment requires a thorough understanding of the ongoing changes in the way businesses organize their activities. A critical insight of the last two decades consists in deconstructing a business into its underlying business processes. A *business process* (**BP**) is "a set of logically related tasks performed to achieve a defined business outcome," see Davenport and Short (1990).

While businesses always faced the task of measuring and monitoring their activities, paper-based information technology (in the form of accounting journals and ledgers) had to rely on pre-filtered and aggregated measures which were typically recorded after a significant time lag. Modern *information technology* (**IT**) utilizes converging computer and networking tools to capture BP information at its source and in the unfiltered and disaggregated form, which makes it possible to measure and monitor business processes at the unprecedented level of detail on the real-time basis.

Continuous auditing (**CA**) is defined as *"a methodology for issuing audit reports simultaneously with, or a short period of time after, the occurrence of the relevant events"* (CICA/AICPA 1999). CA methodology can utilize the IT capability to capture transactional and process data at the source and in the disaggregated and unfiltered form to achieve more efficient, effective and timely audits. An important subset of continuous auditing is the continuous monitoring of business process controls (CMBPC), a task made particularly significant by the passage of Section 404 of the Sarbanes/Oxley Act that requires both managers and auditors to verify controls over the firm's financial reporting processes. The managers' responsibilities are clearly going to be largely based on the work undertaken by the firm's internal audit department.

Kogan et al (1999) discussed the problem of finding a trade-off in the CA implementation between control-oriented and data-oriented CA procedures. There are numerous enterprise environments where process controls are either not automated or their settings are not readily accessible. In such environments, which rely on loosely-coupled legacy data processing systems, automated audit procedures of CA have to be mostly data-oriented (i.e., automated tests of details and analytical procedures), while control testing will involve significant "manual" work.

The tremendous scale and scope of implementations of *enterprise resource planning* (**ERP**) systems since the early nineties has resulted in many companies approaching the state in which their most important BPs are highly automated and fully integrated. This environment of highly automated and tightly-coupled BPs (implemented in integrated enterprise systems) enables the deployment of CA procedures based on continuous monitoring of BP control settings.

Vasarhelyi et al (2004) laid out a series of hypotheses for the implementation of Continuous Audit Systems in such circumstances. They argued that CA would be built on an existing ERP system, implying that it is companies that have already reached full functionality with such systems who would be the first to turn to implementing a CA system as an overlay on their ERP infrastructure. Further, building on the experience with the implementation of ERP systems, as well as the evolutionary path of technology in general, they argued that CA would predictably follow the path of first automating existing manual audit procedures. Once a comfort level with that is reached the implementers would seek to unleash the true productivity benefits of CA by reengineering audit procedures to facilitate continuous auditing, rather than simply taking those procedures as given and making them automatic.

This paper presents a pilot study of implementing CMBPC procedures as a proof of concept in the US internal IT audit department of Siemens Corporation, one of the world's largest transnational companies. It provides an important test bed, using real world audit programs and practicing internal auditors to examine the challenges, constraints and opportunities that face a CA implementation, and the extent to which it fits the implementation model laid out by Vasarhelyi et al (2004). CA has moved from being an academic concept to a state in which CA software is being developed and offered by private industry. If CA is indeed to be the future of auditing, as has long been predicted, then the next step is its implementation for the day to day use of practitioners as opposed to pilot projects led by academics. It is this evolution that this paper examines, deriving important takeaways for the process of implementing CA, both its technological and behavioral aspects. As Alles et al (2002) pointed out, the main constraint on CA is not the supply of technology, but the demand for it, and by extension, the human and economic forces that shape its implementation. Insights into those can only be obtained as a result of actual implementations such as the one reported on in this paper.

We start this paper by first providing in Section 2 a description of the pilot study site and the forces that shaped the implementation. Section 3 discusses the conceptual basis for the continuous monitoring of a business process controls. Section 4 then provides a detailed discussion of the implementation of CMBPC at Siemens. We then discuss the lessons learnt from the pilot study. Section 5 examines the key issue of the difference between automation of pre-existing audit procedures and their reengineering to exploit the full power of CMBPC. One of the major conclusions from this pilot study is that formalizing manual audit procedures to facilitate automation is much more difficult than might have been anticipated, but at the same time, business considerations constrain the ability for clean slate reengineering. Section 6 then considers another important topic that arose from the implementation, the need to carefully manage audit alarms, to balance type I and II errors, while Section 7 discusses the need for an audit trail for the CA system. Section 8 examines the options in the change management process for moving from the pilot study to an industrial strength software application, while Section 8 offers concluding comments.

2. The Pilot Implementation

The implementation took place at the IT Internal Audit department of the US region of Siemens Corporation, which provides internal IT audit services for some 70,000 US employees, generating $20 billion dollars in annual sales across a variety of business sectors. A global characteristic of this company is that it is heavily ERP and SAP R/3 centric.

Siemens approached the authors to conduct an implementation pilot of CA at their site. This presumably only adds to the odds of success, as opposed to when academics approach companies with the desire to conduct an experiment of this sort. What is of greater interest is the precise aims of this IT Audit Department and Siemens for this project. The main motivation was cost savings through greater productivity. Figure 1 is taken from a presentation prepared for internal and external audiences by Siemens' Internal Audit to explain why the project was undertaken. While the projected cost savings have yet to be tabulated and compared against this projection, Figure 1 is interesting in its own right for demonstrating both the value proposition driving actual CA implementation and for indicating the kinds of cost savings it can bring about. Even achieving a fraction of these projections would give this project a very high ROI.

Siemens has SAP installations spread throughout the United States that need to be audited on a regular basis.. The SAP IT audit process is comprehensive across major SAP modules, is performed online, but essentially manual and obviously episodic. The end to end process takes nearly 70 person days for a single SAP system and involves a great deal of traveling by the audit staff. The ability to automate some audit checks was considered to potentially lead to large cost savings, even leaving aside any increase in effectiveness. An additional key consideration was the anticipated demands of implementing Section 404 of the then recently passed Sarbanes/Oxley Act. In a tight economy, the challenge was to cope with the additional burden of this act while not adding significantly to headcount in the internal audit department. Siemens also desires to expand the scope of SAP audit to cover additional and new SAP modules and functionality without expanding the time needed to complete the audit. CA was seen as a potential tool, if not used directly in Section 404 work, then by reducing the existing workload on the audit team which could then be redeployed to Section 404 tasks. The best payoff, of course, was if the CA system was also ultimately able to contribute to meeting the 404 needs.

The scope of the audit was largely determined by a set of what we will call "*Audit Action Sheets*" (**AAS**s) that were created for the guidance of the internal audit department by Siemens's with support from their external auditors (Figure 2 presents an "anonymized" version of an AAS). The external auditor is one of the Big Four public accounting firms. There were several hundred AASs that usually included a mixture of tasks and procedures covering mostly configurable process controls common to any SAP application. Some of them could only be accomplished by a human, such as interviewing the client about their reconciliation procedures, while some others involved well-scripted interactions with the client's enterprise system which were broadly along the lines of the following easily automatable procedure:

- *Execute a certain SAP R/3 transaction and/or report and verify that its result is as specified.*

Additionally, there were certain other procedures which execution seemed easy to automate, but which presented a challenge in automating the evaluation of their results, such as in the following example:

- *Retrieve and examine the list of users who have the "administrator" access privileges to a particular system, and determine whether this level of privileges is appropriate for everybody on the list.*

The results of each AAS are graded by the auditor on an ordinal scale, with the resulting scores then aggregated across the AASs to obtain an overall score for a site. The overall evaluation and degree of compliance of each site given by the internal auditor depends on that total score.

Importantly, the external auditor was willing to partially rely on the work done by the internal auditor for year end and Sarbanes/Oxley 404 compliance subject to the AASs being followed. For this reason, it was very important that the CA system followed as closely as possible the AASs, an approach which has both pros and cons as the research group was to find out. In other words, the pilot study confirmed the Vasarhelyi et al (2004) hypothesis that CA would first automate existing audit procedures rather than reengineer them to better suit the needs of the CA system.

In particular, there was a subset of AASs that the internal auditors already examine off site, usually before traveling to the site. The internal auditors obtained the data relating to these AASs from Siemens' SAP system and applied the AASs to that data. Thus Siemens had already determined which of the AASs were best suited for this kind of remote inspection and the research team did not have to take up that issue. However, this subset of AASs was still being completed manually, even if off site. The first task of the CA project was to see if at least some of these off site AASs could be done automatically.

The implementation of the CA pilot at Siemens involved the following steps.

1. Determining the best mode for the continuous monitoring of the chosen BP controls.
2. Developing system architecture for this task, whether by using a monitoring and control layer or some sort of embedded audit module.
3. Determining the interaction and integration between the CA mechanism and the ERP system.
4. Developing guidelines for the formalization of the AASs into a computer executable format. In particular, determining which AASs are automatable (formalizable) and which require reengineering.
5. Creating processes for managing the alarms generated by the automated CA system and putting in place the required set of audit trails.
6. Formulating a change management plan to move the project from the pilot stage to industrial strength software.

The research team consisted of several faculty members and several doctoral students and research assistants. They worked closely with senior internal auditors at Siemens, including the head of the internal IT audit department. The participation of the latter was essential when formalizing the AASs, both for resolving the inevitable ambiguities and uncertainties and for validating the formalized versions of the AASs by the auditors. In addition, a doctoral student observed a site audit by the auditors to determine the way in which the AASs were executed in practice.

3. Continuous Monitoring of Business Process Controls

Kogan et al (1999) and Vasarhelyi et al (2004) put forward hypotheses about how CA would be implemented. From the pilot study we obtained a far more nuanced view of the drivers, constraints and the most productive approaches towards implementing CA in practice. The lessons we have learned are presented in the form of a conceptual model for designing the system for continuous monitoring of a business process controls, as depicted in Figure 3.

3.1. Modes of CMBPC

Continuous monitoring of BP controls relies on automatic procedures, and therefore presumes that both the controls themselves and the monitoring procedures are formal or formalizable. Note that the latter is necessarily premised on the former. Formalization of BP controls, while important in its own right, has been precipitated by ERP implementations and the ongoing Section 404 of Sarbanes-Oxley compliance work.

The verification of existence, correctness and functioning of BP controls can be accomplished in three different ways:

- Firstly, one can observe a BP and verify if the observations agree with the proposition that a control exists, is correct and functioning. The benefit of this approach is that it can be applied even in those environments in which controls are not directly accessible by the auditor. The problem with this approach is that the observed behavior of the BP may not completely cover the whole range of situations in which the control is expected to function, and therefore there is no assurance that this control will be functioning as expected under all circumstances.
- Secondly, in the case of preventive controls, one can attempt to execute a prohibited BP behavior (e.g., run a prohibited transaction such as recording a large purchase order without proper authorization) to verify that such behavior cannot happen. In the case of detective or compensating controls, the auditor can verify that the prohibited behavior is detected and compensated for. While such control testing provides much stronger evidence than the previous approach, it is highly unlikely that an auditor (even an internal one) will be allowed to execute such type of "penetration testing" on a production enterprise system. Under most common circumstances, the best an auditor can count on is the read-only access to the production system.
- Finally, one can retrieve the control settings stored in the enterprise system and verify that they match the benchmark. The benefit of this approach is that it requires just the read-only access to the enterprise system and provides a very strong evidence since it actually confirms that the control is indeed what it has to be. The critical assumption in this approach is that the programming code of the

control in the production enterprise system is correct, since what is verified in this approach is only the control settings. This assumption seems to be reasonable with respect to the standard controls built into modern packaged ERP systems such as SAP R/3 or Oracle Applications. However, an ERP system can be customized, and in the case of customized controls additional initial control verification work may be needed to complement the ongoing monitoring of BP control settings.

The analysis above implies that in the case of highly integrated and standardized enterprise system environments, the most appropriate approach to CMBPC is to implement continuous monitoring of BP control settings. Modern ERP systems make their automated BP control settings accessible online from the CA system. The process of monitoring itself falls within the general CA framework develop in Vasarhelyi et al (2004) of obtaining assurance by continuously comparing the actual observations (in this case the control settings) against the benchmarks. Therefore, the determination of the appropriate benchmarks for the acceptable BP control settings constitutes a critical part of implementing a CA system. Clearly, such benchmarks are often enterprise-dependent. In the case of large multi-national companies certain control setting benchmarks may depend on the country or a particular unit of an enterprise, which will complicate the setup of the CA system.

A critical parameter in the CA system is the frequency (e.g., daily, hourly) of comparison of the actual BP control settings with the benchmarks. This is a generic issue in any CA system setup, and the optimal frequency may depend on many different features of the environment and the controls under consideration. Note that while higher frequency is indeed beneficial for achieving higher levels of assurance (since less time is available for undesirable adjustments or malfeasant transactions), the main problem with the excessive frequency is not the processing capability of the CA system, but rather the performance penalty imposed by such queries on the production enterprise system. While an hourly frequency will usually not present a problem, hitting a production system every second with a query to retrieve voluminous control settings may be problematic, especially during the working hours. A bypass of this problem, as described by Vasarhelyi and Halper (1991) is the utilization of reports that are as a matter of course prepared by corporate IT.

The main task of a CA system is to take action in case the observed BP control values deviate from the benchmarks. We call such deviations **exceptions**. A CA system has to automatically generate alarms in case of critical exceptions, such as individual accounts without passwords, or in case if numerous non-critical exceptions result in the aggregation of weaknesses in certain control areas (e.g., segregation of duties). The alarms are always sent to the auditors, and can optionally be sent to responsible enterprise personnel and/or enterprise managers, as well as other relevant parties.

3.2. System Architecture for Continuous Monitoring of BP Controls

The design of a system architecture for continuous monitoring of BP controls can be based either on an independent system usually called the *monitoring and control layer* (**MCL**), see Vasarhelyi et al (2004), or on a subsystem of an enterprise system usually called the *embedded audit module* (**EAM**), see Groomer and Murphy (1989). While in theory, the actual CMBPC system can utilize a combination of these two approaches, to understand clearly their relative advantages and disadvantages, one should analyze them separately.

MCL is implemented in a separate computer system, which is usually owned and operated by the auditor. In many cases, the MCL system will not even share premises with the enterprise system, and will rely on remote (read-only) access to the enterprise system at the application layer. This, along with taking a broad extraction of controls data, is why the code and environment of MCL can be well-protected, and the enterprise data retrieved by MCL can be presumed to be absolutely safe and not susceptible to pre or post extraction manipulation by the enterprise personnel (even by those who have the super-user privileges). On the other hand, as was mentioned in the previous section, MCL cannot query the enterprise system too often, and therefore can miss suspicious enterprise events.

EAMs by their nature are tightly coupled with the enterprise system. They can even be provided by the ERP system vendors as standard parts of the system. Among the advantages of this architecture is the independence of EAMs of the availability of network connectivity or bandwidth, and easier access to voluminous enterprise data. The most essential advantage is that EAMs can be implemented as triggers flipped by suspicious business events, which eliminates the need for large or high frequency queries to assure that such an event is caught and analyzed in real time, thus preventing the possibility of a cover-up between the queries. However, EAMs are intrinsically more vulnerable to manipulation, especially by the enterprise personnel who have the super-user privileges. Neither the code of EAMs nor the results of their processing are completely safe, and safeguarding them will require some very innovative and complicated cryptographic techniques, well beyond the range of those which are currently utilized in practice.

Another critical advantage of MCL over EAMs is that the implementation of MCL is less reliant on the cooperation of the enterprise personnel. Not much is required from the enterprise personnel beyond granting read-only access to the system. On the other hand, the implementation of EAMs, especially if not provided by the ERP vendor itself, requires the participation of the enterprise personnel in a complicated development and customization process to incorporate EAMs into a fully tested production version of the enterprise system. Such level of cooperation is difficult to obtain, and this issue dims the prospects of EAMs, at least in the foreseeable future (until the ERP vendors incorporate fully developed CMBPC functionality into their products).

3.3. Interaction of MC Layer with the ERP System

Modern integrated enterprise information systems have a 3-tier architecture consisting of the presentation, application, and database layers. Each of these layers is typically run on a separate computer system. While the database layer contains all the enterprise data, all the business logic is coded and executed in the application layer. This 3-tier enterprise system architecture creates a dilemma of whether MCL should interact with (or EAM should reside in) the application or database tier of the enterprise system.

MCL can query the enterprise system through the application tier using its application program interfaces (e.g., BAPIs in the case of SAP R/3). This approach is usually well-supported by system vendors and the APIs are well-documented. Analogously, an EAM can be implemented as a sub-module of the application (e.g., coded in ABAP in the case of SAP R/3). While this is more laborious and prone to problems discussed in the previous section, this approach is also well-supported and documented by the enterprise system vendors.

MCL can query the enterprise database directly (using SQL through ODBC). While in principle this approach is more versatile than querying through the application tier since it is not constrained by the structure of the enterprise business objects, in reality the schemas of enterprise databases are so complex and enormous (they are highly normalized and contain upwards of 20,000 tables) that digging out anything which is a not a well-documented business object is close to impossible. Analogously, EAM can be implemented as a trigger (written in SQL) stored in the database. However, using triggers in transactional databases will have an adverse effect on the database performance, in some cases slowing down the enterprise transaction processing system to a standstill.

The latter approach is strongly resented by enterprise personnel and (in the case of EAM) is de facto prohibited by enterprise software vendors. Therefore, only the former approach can be utilized in implementing CMBPC.

4. Implementing CMBPC at Siemens

4.1. Selection of AASs

Figure 4 illustrates the existing audit procedure utilized by Internal Audit at Siemens. Data was extracted in batch mode from the SAP system that was currently under audit by a proprietary tool know as E-Audit. Its output was a text file which internal auditors would manually examine when completing the AASs that had been assigned to them. E-Audit was also the basis of the CMBP tool, with the goal of making at least some of the AASs automatable.

Data selected for the model was taken from the formal Siemens SAP audit process in the Basis area (the application layer operating system for SAP) covering the application level controls applicable to any SAP system:

CONTROL	**AAS #**
Basic password settings	1.02.000
Password rules and SNC	1.02.010
Handling initial passwords of inactive users	1.02.020
Users in clients 000 and 001	1.02.030
Initial passwords for standard users	1.02.040

System parameters for SAP*	1.02.050
Standard user SAPCPIC	1.02.060
Analyze emergency user concept	1.02.100
System administration	1.02.110
System admin./completeness verification	1.02.120
System parameter settings	1.02.130
User authentication documentation	1.02.999

Under each control review there are one to five control elements reviewed. The twelve sheets selected represent 5% of the population of audit action sheets in the Siemens program, but the Basis section selected (section 1.02.XXX) is representative of the population of data in terms of its applicability to continuous auditing or assurance. Throughout the SAP audit, the auditor is instructed to perform a variety of review procedures on the SAP application. These range from very simple checks of standard system parameters to securing more subjective data requiring input from interviews with key IT personnel or business users. In some instances the pilot model could not evaluate the audit control described on the sheets because additional input is required from an interview.

As an example, the sheet "Analyze emergency user concept" (1.02.100) requires the auditor to check empirical variables on the SAP ERP system related to the use of passwords for emergencies. It also requires the auditor to interview the client to gain an understanding of methodology and risk based strategy behind the authorization concept the company uses. Such additional information cannot be readily incorporated into an intelligent software model without adding significant complexity and effectively capturing management's thinking process. For the model used in this research work, application controls which required significant formalization and management interviews for gathering input were eliminated from the study.

Another key evaluation criterion common to all the source audit sheets was the process of scoring the sheets on a zero to four scale (zero = no control in place / four = full control) once the appropriate data was selected and evaluated by the intelligent continuous audit analyzer (CA Analyzer) component of the system. The challenge was that the scoring criterion described by the audit action sheets was often ambiguous or vague extending considerable license to the traditional auditor to score the findings between a zero and a four. This was an issue even where the variables evaluated were pass or fail decisions, because there were several variables of unequal importance assigned to one score and some type of weighting was needed to reach a single numeric score. The audit action sheets provide general guidelines for the auditor to reconcile the weighting of scores. This is acceptable with a manual audit process, but problematic when programming intelligent software. The following section from audit action sheet number 1.02.000 used in the pilot model illustrates the issue.

> Rating Criterion:
>
> The RSPFPAR report lists all basic system parameters for password creation:
>
> 1) login/min_password_lng (minimum password length has to be 8 characters)
>
> 2) login/password_expiration_time (password has to expire after a maximum of 90 days)
>
> 3) login/fails_to_sessions_end (is the number of illegal login attempts before the session is aborted set to 3?)
>
> 4) login/fails_to_user_lock (the number of failed login attempts before system lockout should be set to a maximum of 5)
>
> 5) login/failed_user_auto_unlock (is a system lockout automatically cancelled overnight?; recommended setting = 0)
>
> Rating notes:
>
> Inadequate protection for SAP access (authentication problem) may be provided internally by company staff or by external parties to whom network access has previously been granted. If the IS Guide is not followed, the rating should be (0) = very significant non-compliance. If the respective parameters (see above) have the recommended settings, the rating should be (4) = no non-compliance. In the case of partial compliance, depending on the settings made, rate the audit action sheet (2) = non-compliance.

The rating criterion section of the above audit action sheet is clear, outlining five criteria for password structure with specific variables defined. The rating notes, or scoring instructions, however, are not adequately formalized to allow for programming into intelligent software. The scoring is clear if all or none of the five criteria are met, but if one or two of the variables are not properly set, the auditor would need to make a subjective judgment as to what is the appropriate score. The above example is symptomatic of almost all the scoring criteria used in the model data and typical of most manual audit program scoring models.

Formalization of the above scoring criterion would require some type of risk-based weighting. If all password criteria do not carry equal risk or the definition is subjective as to when a score of one, two, or three is appropriate, then a formal way of combining audit evidence will be required. This formalization process, needed to allow intelligent software to act methodically on this data, forces the auditor to further analyze the risks and priorities of audited variables. We defer further discussion of the formalization issue to section 5 below and now turn to data issues in the CMBPC implementation.

4.2. Data issues

The data based used for the pilot model was stored in a simple MS Access database, quite capable of handling the volume of data utilized in the pilot model. In a production application, however, the size of the database could be significant depending on the size of the extraction supporting the audit plan. The number of ERP systems being evaluated, the length of retention of the data, and the frequency of downloads could all add significant database size and increase the complexity to the continuous auditing / assurance model.

The full download of the E-audit component of the Siemens SAP audit used in the pilot consumed about three megabytes of data per download. This is not significant for a single audit on a single application for a single download. Consider, however, an application where there are 100 SAP ERP systems being evaluated by the continuous auditing analyzer every ten seconds with a data retention requirement of 24 hours. The resulting potential database volume needed to support this application would be a minimum of 2.6 gigabyte of storage. If any of the above variables are increased for a specific application the database capacity requirements could grow very quickly requiring a robust relational database with appropriate database management capabilities and support (such as Oracle or MS SQL Server).

While any of the variables impacting the required size of the database can be adjusted to reduce database size and load, the very nature of continuous auditing / assurance applications indicates the need for a significant short-term storage capacity. The greatest opportunity for reducing database load is probably in adjusting the retention requirements of the system. Data extracted for continuous auditing may only need to be retained if exceptions resulting in alarms which require follow-up are identified by the intelligent software. The logic here is that due to the continuous nature of the review (at or close to real-time) there is no reason to retain data that has passed the evaluation and does not require subsequent action. Additionally, the retention of sensitive data in a continuous auditing database any longer than necessary carries potential security and confidentiality risks.

Certain types of data, however, will need to be retained in a database for longer periods of time because of time series length requirements for analysis. Accounts payable or receivables data, for example, where the system may need to match invoices or orders with receipts would require time series data. Any data stream seeking to identify a trend will need to be retained in some format at least until the trend is established and documented.

Regardless of the scope, frequency, or nature of the data retention requirements in a continuous audit / assurance process, it is advisable that a robust relational database be used to manage the potential for storing and handling large amounts of data. A good relational database package may provide core or supplemental reporting capabilities. The database application can serve as or in support of the continuous auditing / assurance model's analysis or reporting engine.

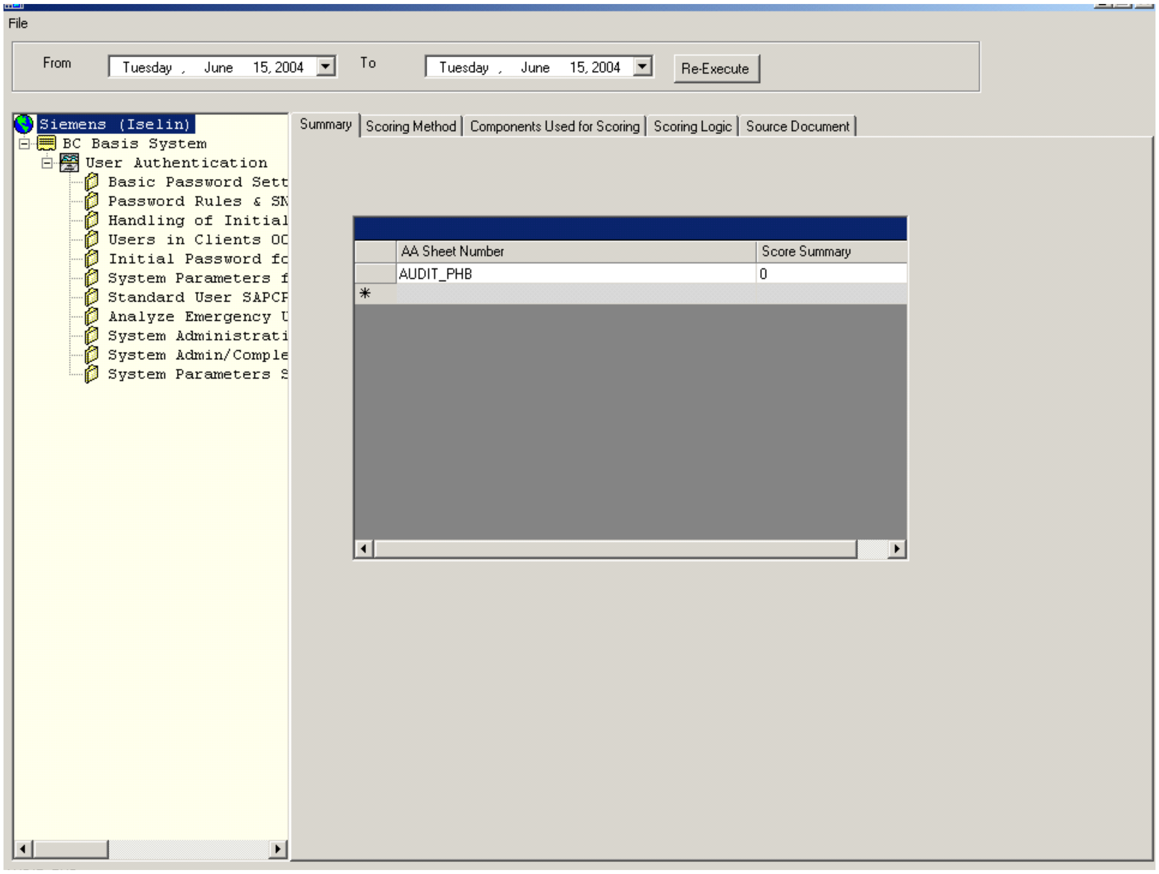
4.3. CMBP coding

The actual software application developed to implement the pilot model showed significant flexibility in addressing the diversity found within the Siemens audit actions sheets as part of the SAP audit process. As described earlier, a subset of actual control data from the E-audit download provided sample data for the model. This data was reviewed against business rules defined from the Audit Action Sheets in the CA Analyzer to identify exceptions and report alarms to the appropriate compliance personnel. Figure 5 illustrates the process structure with multiple SAP / ERP systems, a relational database, a CA Analyzer (intelligence) and an alarming or workflow process.

The focus of the pilot model was the CA Analyzer section. The common SAP systems (at least in terms of versions and basic functionality) and the E-audit download already exist in the SAP audit process at Siemens. The pilot model was developed in Visual

Basic to serve as a test environment for evaluating technical research questions regarding continuous auditing / assurance. Visual Basic provides excellent development and research results, but may not be robust enough to handle large data volumes in continuous auditing applications. The screen shots and comments outline the basic functionality and key aspects of the pilot model highlighting the leverage and limitations of the proof of concept model in evaluating continuous audit / assurance research.

The summary screenshot provides a list of the business rules evaluated in the model.



From Tuesday , June 15, 2004 To Tuesday , June 15, 2004

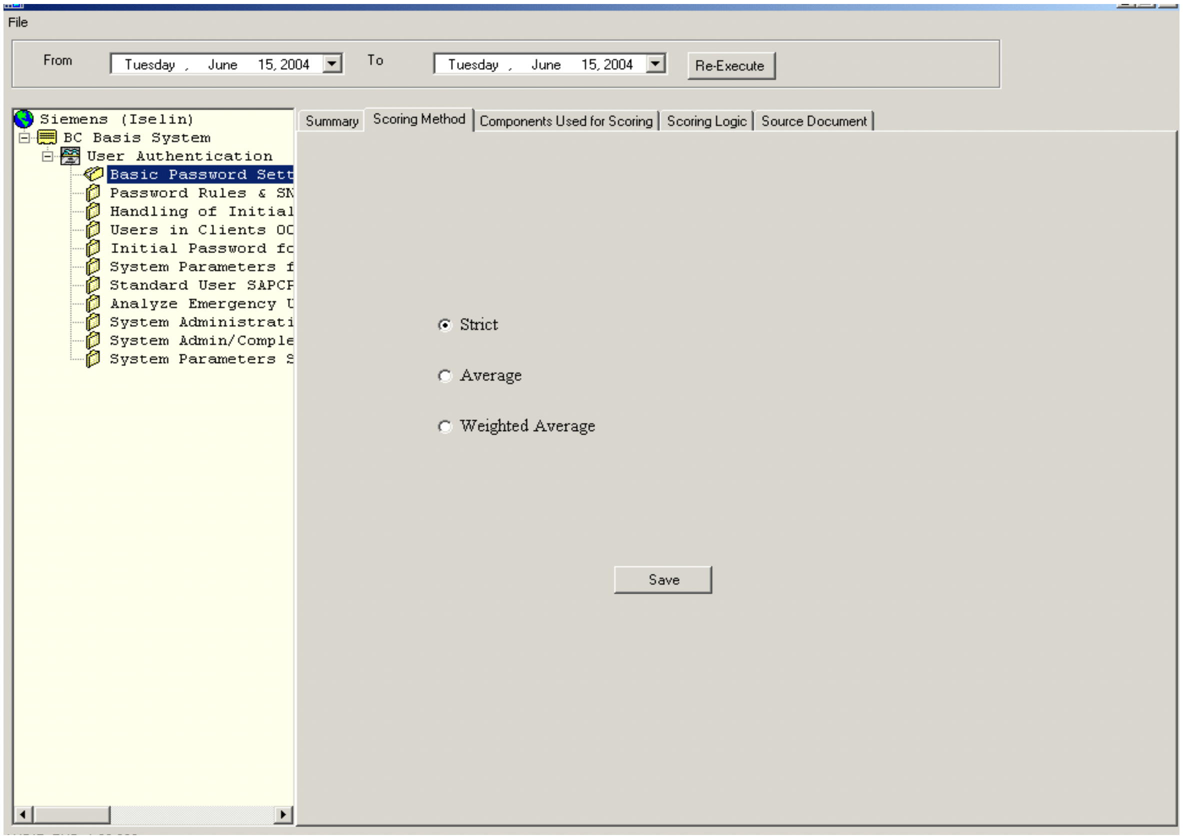
	AA Sheet Number	Score Summary
	AUDIT_PHB	0
*		

These can be modified, within certain parameters, by the user allowing for rule changes to be made without programmer intervention. This idea, discussed above, of needing system agility to create or change business rules without the cost and inflexibility of coding changes is a critical component of a successful continuous auditing application. Writing flexible scripting software is difficult, and while there are many business rule scripting software solutions out on the market, most are specialized for select applications. With any scripting software, developed or off the shelf, there is always a tradeoff between flexibility and complexity.

Note the summary screen identifies the data range for the selected data. This is critical to assuring the data is meaningful for certain types of data. Latency and range determine if an identified exception in the data set is meaningful and should generate an alarm. For example, if overdue accounts payable invoices are being reviewed, the selected data is only meaningful if it provides invoices and cash receipts documentation within the appropriate date ranges to be considered overdue. The date ranges will also be important in determining when data should be purged from the system to avoid security or capacity issues within the system.

Requirements such as these expand the required amount of data that must be stored in the relational database feeding the continuous auditing analyzer and the complexity of data purging methodology in the system. The next several screens in the pilot model address the interpretation and scoring of the respected audit steps or audit action sheets. This is critical because as the sample AAS discussed above demonstrates, each audit step may contain five or more variables to be checked resulting in a common score for a selected audit step.

In that example, five variables must be combined to determine a score for this one audit action sheet. Each authentication parameter may carry a different degree of risk or exposure for the audited organization and therefore require some type of weighting to arrive at an accurate score in the zero to four scoring range of the audit model. The scoring instructions in the manual audit plan, from which the pilot model was derived, provide considerable latitude to the auditor in subjectively weighing the individual scores based on risk to arrive at a cumulative score. As discussed above, an automated model does not extend this level of flexibility to the users and requires much more formalization to be implemented in algorithms. To address this issue the model provides the following options for summarizing multiple elements within a single audit step:



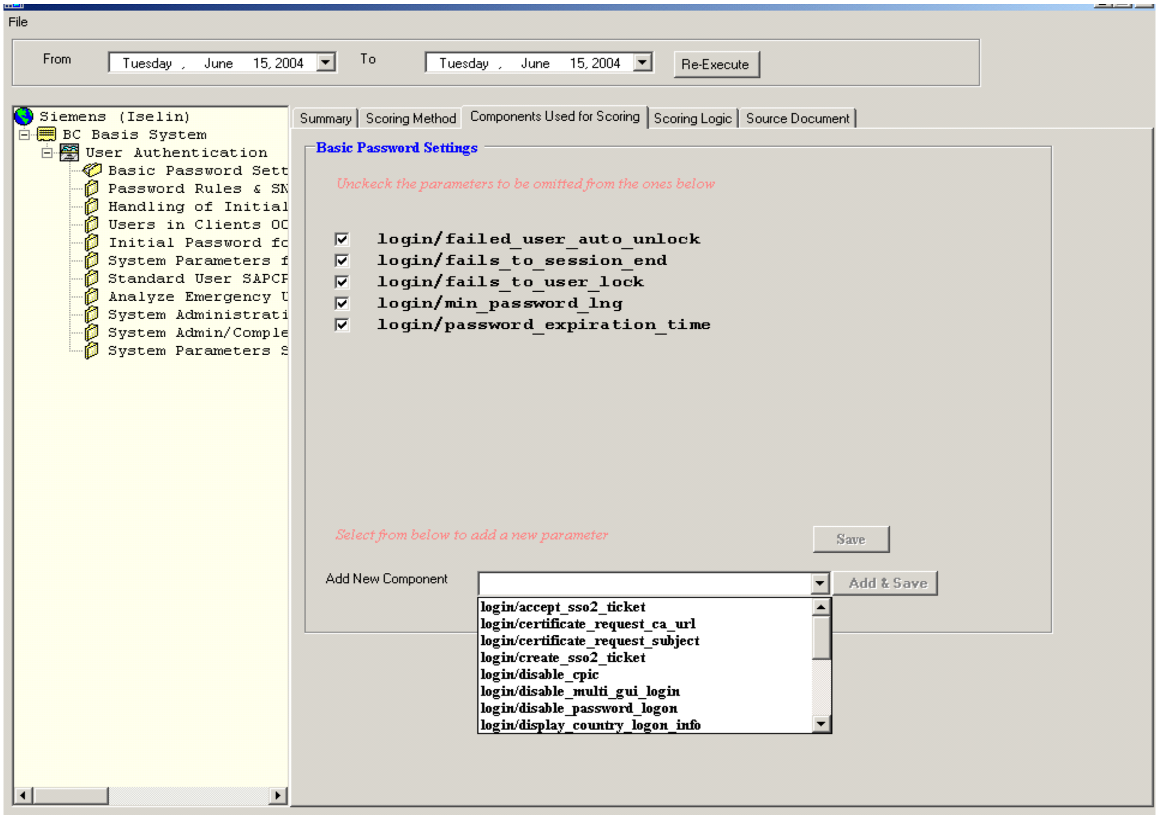
From Tuesday , June 15, 2004 To Tuesday , June 15, 2004

Siemens (Iselin)
BC Basis System
User Authentication
Basic Password Sett
Password Rules & SN
Handling of Initial
Users in Clients 0C
Initial Password fc
System Parameters f
Standard User SAPCF
Analyze Emergency U
System Administrati
System Admin/Comple
System Parameters S

● Strict
○ Average
○ Weighted Average

As indicated, a strict, average, or weighted average scoring can be used. So if a particular outcome on a single variable determined the overall score, a strict weighting selection would be appropriate. If each of five variables carried an equal risk weighting, an average weighting would be used. If, however, the minimum password length control element, for example, was deemed much more important than the other variables, it would be appropriate to use the weighted average scoring method. This formalization is critical in ensuring that measures and scoring are standardized and that the continuous auditing model is scalable and repeatable across SAP platforms and audited organizations.

The variables to be included in the review of a particular audit action sheet are shown in the following screen under the tab "Components used for scoring." This simply allows the auditor or operator to define the control elements to be included in the evaluation and scoring – the scoring method or weighting is applied to all selected components. The input box at the bottom allows the operator to add or delete elements as needed, providing some of the required agility in the system.



From Tuesday , June 15, 2004 To Tuesday , June 15, 2004

Siemens (Iselin)
BC Basis System
User Authentication
Basic Password Sett
Password Rules & SN
Handling of Initial
Users in Clients 00
Initial Password fc
System Parameters f
Standard User SAPCF
Analyze Emergency U
System Administrati
System Admin/Comple
System Parameters S

Basic Password Settings

Unckeck the parameters to be omitted from the ones below

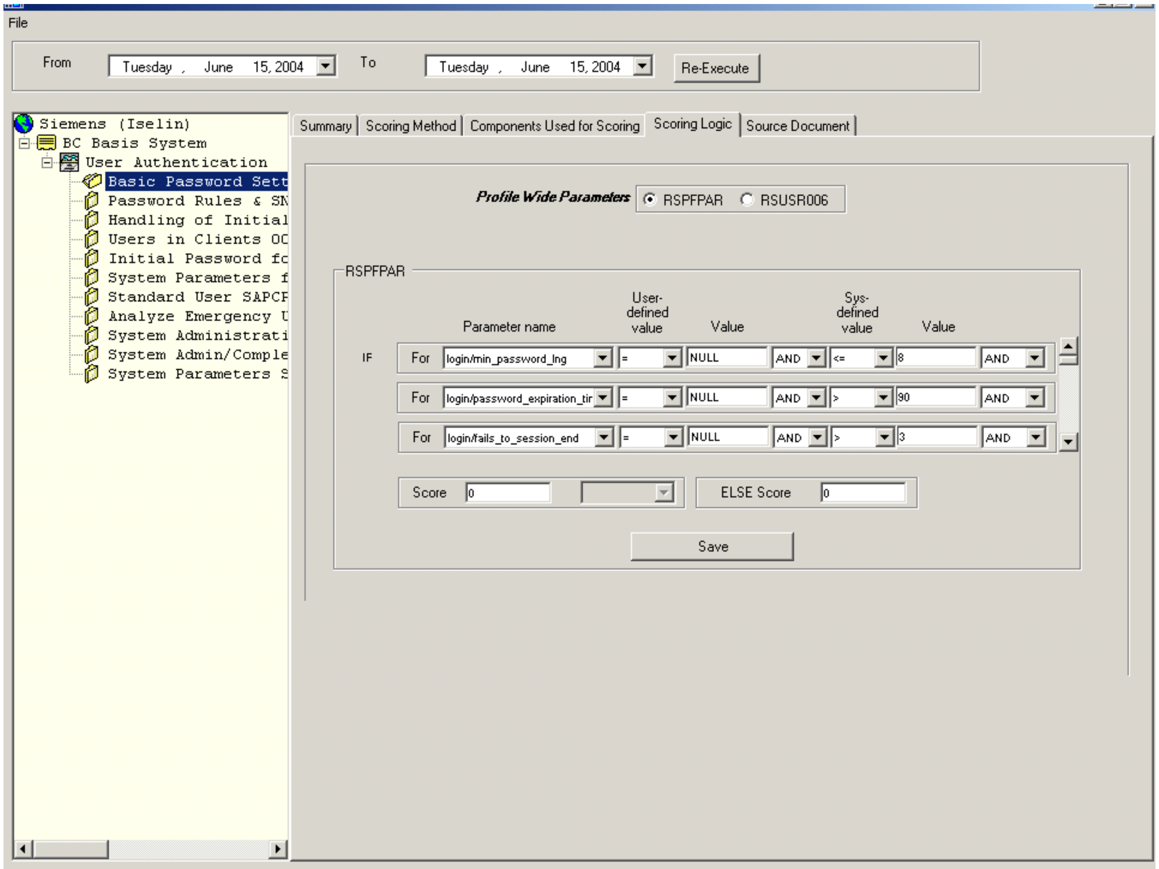
☑ login/failed_user_auto_unlock
☑ login/fails_to_session_end
☑ login/fails_to_user_lock
☑ login/min_password_lng
☑ login/password_expiration_time

Select from below to add a new parameter

Add New Component

login/accept_sso2_ticket
login/certificate_request_ca_url
login/certificate_request_subject
login/create_sso2_ticket
login/disable_cpic
login/disable_multi_gui_login
login/disable_password_logon
login/display_country_logon_info

The next screen, also related to the scoring methodology, provides an example of a scripting model allowing the user to define a specific criterion for a score. This provides flexibility for the auditor to further formalize the evaluation and scoring process without making hard-coded programming changes.



From Tuesday , June 15, 2004 To Tuesday , June 15, 2004

Profile Wide Parameters ● RSPFPAR ○ RSUSR006

RSPFPAR

		Parameter name	User-defined value	Value		Sys-defined value	Value	
IF	For	login/min_password_lng	=	NULL	AND	<=	8	AND
	For	login/password_expiration_tir	=	NULL	AND	>	90	AND
	For	login/fails_to_session_end	=	NULL	AND	>	3	AND

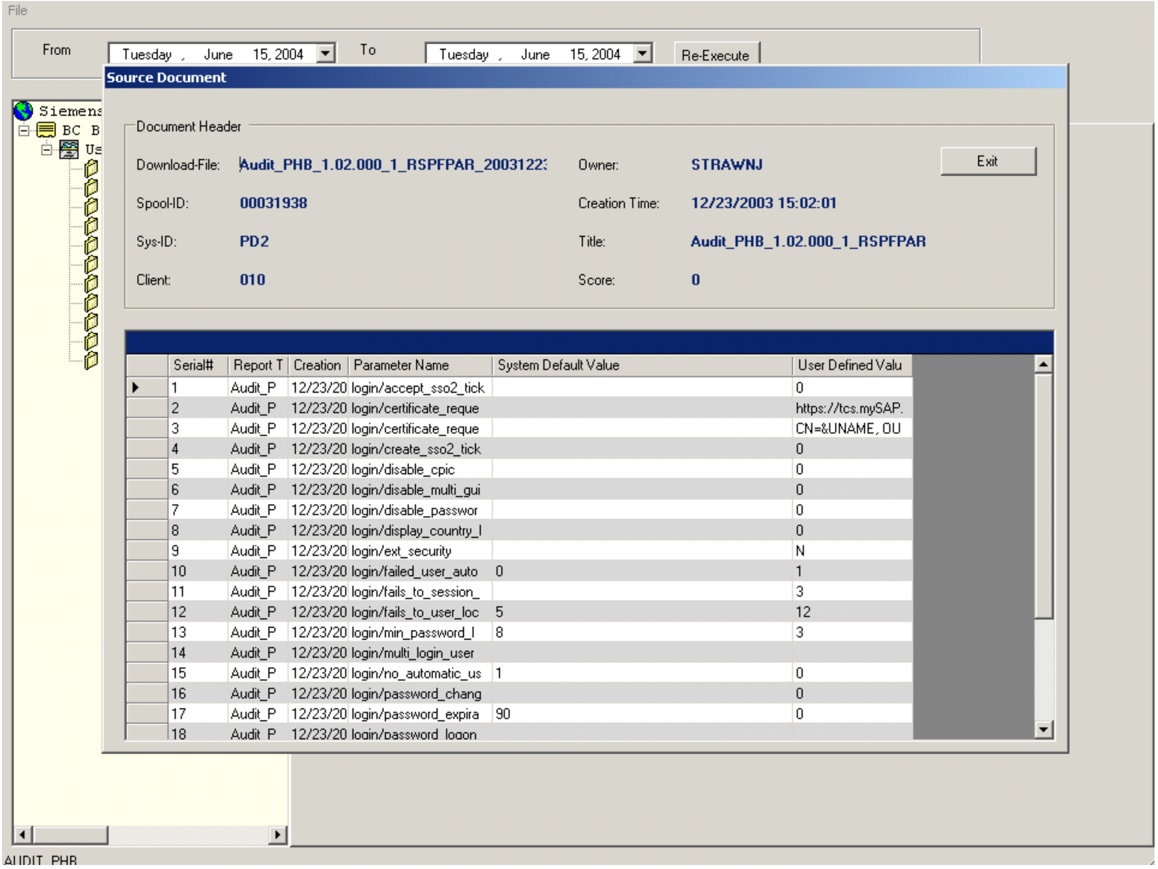
Score 0 ELSE Score 0

In the example above, the report RSPFPAR reviews three control elements and determines a specific score based on the outcomes. The measured criteria include:

- Checking that the password length is 8 characters or more
- Checking that the password expiration is set to 90 days or less
- Checking that the allowable attempted logins is three or less.

If all of the above criteria are met, a predetermined score will be assigned. If the criteria for any of the variables are not met, a single alternate score will be assigned. This is an example of a strict criterion mentioned above with no weighting or averaging performed. There are many control reviews falling into this category where a combination of events results in a passing score and an exception in any event results in a failure of the entire review. Despite the flexibility provided by algorithms like the above, many control elements or monitoring schemas require much more complex logic which would require extensive custom programming thereby limiting the agility of the software.

The final tab in the pilot model provides a listing and documentation of the source data on which the pilot model acts. This provides an organizational aid to view and manage the inputs to the model. Included are a file name, spool id, system number, owner, and creation date. This information is critical to determining the integrity of the data and the appropriateness of exception or alarms generated by a control check. The date stamp is most critical to assuring the integrity and appropriateness of data.



From: Tuesday , June 15, 2004 To: Tuesday , June 15, 2004

Document Header

Download-File: Audit_PHB_1.02.000_1_RSPFPAR_2003122:

Owner: STRAWNJ

Spool-ID: 00031938

Creation Time: 12/23/2003 15:02:01

Sys-ID: PD2

Title: Audit_PHB_1.02.000_1_RSPFPAR

Client: 010

Score: 0

Serial#	Report T	Creation	Parameter Name	System Default Value	User Defined Valu
1	Audit_P	12/23/20	login/accept_sso2_tick		0
2	Audit_P	12/23/20	login/certificate_reque		https://tcs.mySAP.
3	Audit_P	12/23/20	login/certificate_reque		CN=&UNAME, OU
4	Audit_P	12/23/20	login/create_sso2_tick		0
5	Audit_P	12/23/20	login/disable_cpic		0
6	Audit_P	12/23/20	login/disable_multi_gui		0
7	Audit_P	12/23/20	login/disable_passwor		0
8	Audit_P	12/23/20	login/display_country_l		0
9	Audit_P	12/23/20	login/ext_security		N
10	Audit_P	12/23/20	login/failed_user_auto	0	1
11	Audit_P	12/23/20	login/fails_to_session_		3
12	Audit_P	12/23/20	login/fails_to_user_loc	5	12
13	Audit_P	12/23/20	login/min_password_l	8	3
14	Audit_P	12/23/20	login/multi_login_user		
15	Audit_P	12/23/20	login/no_automatic_us	1	0
16	Audit_P	12/23/20	login/password_chang		0
17	Audit_P	12/23/20	login/password_expira	90	0
18	Audit_P	12/23/20	login/password_logon		

The management and integrity of data used for continuous audit / assurance is critical to the effectiveness of the monitoring system and requires an increasingly mature metadata process as the volume and complexity of the data increases. A large continuous auditing system will need a robust relational database and detailed metadata processes to effectively understand and manage large volumes of data. The pilot model used in this research was limited by the small volume of data used in the study. A large relational database application, such as SAP's Business Warehouse would be an appropriate tool for managing large volumes of data in support of a production ready continuous auditing process. Such database tools include sophisticated metadata and reporting tools. SAP's

Business Warehouse includes a comprehensive reporting tool (BW-Bex), which would allow the auditor to readily search, parse, or purge data from the application.

After this overview of the pilot implementation, we now turn to the lessons learned in the course of its execution.

5. Formalizing and Reengineering the Process of Auditing to Implement CMBPC

Vasarhelyi et al (2004) hypothesized that: *"While the extent of application of CA clearly decreases with the increase in the complexity of the audit object, we argue that certain audit procedures can still be formalized and automated even at the high end of the continuum of audit objects"*. This pilot study provided a means of putting this hypothesis to the test. The question is whether the constraint of working with preexisting AASs will reduce the gains from CA. Efficiency was one of the main drivers for the internal auditors to engage in our pilot study. It is very tempting to attempt the implementation of CMBPC starting with a clean slate, since it allows for the cleanest, most logical and efficient solutions. However, our experience has shown that the messy and busy day-to-day reality of the modern enterprise and their under-resourced internal audit organizations make the ideal clean slate approach impossible. After all, even clearing a completely new internal audit program with the external auditor presents tremendous complications and delays that can kill the whole endeavor. This is the reason why our approach in the pilot study was to stay as close as possible to the approved internal audit program.

Automation requires formalization of audit procedures. Approved audit programs are not highly formalized and most often reflect the legacy of the traditional manual audit / interview approach to auditing. Different human auditors interpret the same program somewhat differently. Our pilot study analysis of the approved internal IT audit program shows that certain parts of the program are formalizable while other parts are not. It is usually possible to separate the formalization of the control testing part of audit program from the formalization of the evaluation of the results of testing. Any CA implementation requires the formalization of both the testing and the evaluation parts of AASs.

Our experience indicates that the most effective and efficient way of formalizing the formalizable parts of the audit program is to have the most experienced audit personnel contribute to this job. It is also important to assure that every formalization is discussed by at least two experienced auditors to uncover and resolve possible ambiguities and diverging interpretations. Research and our experience seem to indicate that more is formalizable than commonly believed. At the same time, significant parts of AAS do lend themselves to formalization and automation (such as those audit procedures that require human observation of BPs and interviewing enterprise personnel).

One of the main findings of our pilot study is the fact that even if one attempts to keep the existing audit process as intact as possible, implementing MCL requires reengineering this process. The unavoidability of audit process reengineering stems from the necessity for the formalizable and non-formalizable parts of the audit program to be identified and handled separately. In our implementation we proposed that the formalizable procedures to be separated from non-formalizable ones, then automated and executed with high frequency (continuously), while non-formalizable procedures should continue to be done manually.

Formalization is beneficial for many reasons. Not only it is a prerequisite for AAS automation, but also it creates uniformity and makes it easier to assure that every formalized and implemented procedure is state of the art. The process of re-engineering controls by experienced auditor for purposes of formalization of controls leverages automation to produce more efficient methods of addressing key controls. As has been mentioned before, high frequency of execution of automatic procedures provides higher levels of assurance.

6. Control and Alarm Hierarchy and Its Management in CMBPC

The exceedingly large scale and scope of modern enterprise controls make their organization particularly important. The standard approach to organizing controls starts with identifying various risk areas, breaking them down into sub-areas, and then developing controls to eliminate or mitigate these risks. The resulting system of enterprise controls forms a top-down hierarchy. In the case of a highly automated and integrated enterprise information system, the hierarchy of IT controls reflects the structure of the enterprise system. In the example of SAP R/3, the top level of the IT control hierarchy corresponds to the main components of the system such as the basis system, financial accounting, materials management, etc.

Even in the best run enterprise, the on-going control monitoring process is expected to identify certain control violations. The severity of such violations (called exceptions) can differ significantly, from critical exceptions, such as non-existing or default super-user passwords, to relatively benign ones, such as a regular user password which is older than three months. Developing a sound scientific approach towards the measurement of severity of various exceptions presents a challenging research problem. The empirical approach, taken in our pilot study following the approved internal IT audit program, assigns a numerical score reflecting the severity of the exception (with "4" representing the critical failure and "0" corresponding to the perfectly functioning control).

One interesting finding of our implementation is that in many cases in which the testing part of the audit procedure is relatively easily formalizable, the formalization of the evaluation part often presents a significant challenge. Not surprisingly, while there is usually no disagreement about assigning the scores of 0 or 4, the choice between the intermediate scores of 1, 2, and 3 is often controversial, reflecting the lack of the sound measurement methodology and the ambiguity of whether the scale of measurement is ordinal, interval or ratio (with ordinal usually being the most, while ratio the least appropriate).

The assessment of the state of enterprise controls provided by the CMBPC system in the form of exception evaluation scores is created not only for informing the auditor who logs into the CA system, but even more importantly, it provides a set of actionable items to precipitate corrective measures for identified control deficiencies. Following the approach developed in the CA literature, see Vasarhelyi and Halpern (1991) and Vasarhelyi et al (2004), we implement the active component of the CMBPC system as automatically triggered audit alarms. One of the critical design decisions in implementing CMBPC is the choice of exception conditions to trigger alarms. Clearly, critical exceptions (having the score of 4) represent one type of such exception conditions. Additionally, the accumulation of non-critical exceptions in certain control areas (such access control or segregation of duties) also represents the conditions necessitating the generation of alarms.

Our pilot study allowed us to identify an important practical and theoretical problem associated with the automatic generation of alarms in CMBPC – the alarm flood. While alarming is critically important in CMBPC since it makes it possible to correct the identified problems in close to real time, if the number of alarms generated by the CMBPC system explodes, then it will hamper the ability of auditors and other enterprise personnel to react and correct the identified problems. The worst case outcome of the alarm flood happens if the enterprise personnel decide to ignore the CMBPC system alarms altogether, or if the auditor is forced to switch all the CMBPC system alarming off. This problem is a particular case of information overload first discussed as a cognitive effect of CA by Kogan et al (1999). The highest likelihood for the alarm flood to occur is during the ramp-up period right after the CMBPC system goes live.

Transition from the manual audit process to CMBPC and automatic alarms can result in the alarm flood for two different reasons. Firstly, it is possible that the enterprise system has a lot of sub-optimal BP control settings, and those have to be corrected. However, this initial correction can take a significant amount of time, and the adopted production configuration of the CMBPC system will require sending alarms over and over again. The initial flood of alarms has to be anticipated, and to prevent it, the "go live" configuration of the CMBPC system has to limit the number of alarms initially, and then broaden the alarm conditions gradually. The second reason the alarm flood can happen is if the configuration of the CMBPC system is overly conservative (which is not unlikely given the common personality traits of most auditors), and a lot of alarms result from exceptions which can be viewed as tolerable (such as the purchase order authorization threshold slightly exceeding the corporate policy limit). To prevent this problem from happening, the possibility of the alarm flood has to be taken into consideration while determining the production configuration of the CMBPC system.

The ongoing maintenance of the configuration of individual alarm conditions is a laborious process, which may require an inordinate amount of auditor's time if the CMBPC system is not designed to alleviate this problem. After the production configuration of CMBPC is put in place, the most commonly required maintenance is changing the recipients and enabling/disabling certain alarms. The fundamental reason for the possibility to simplify alarm management is due to the fact that the alarms form a hierarchy corresponding to and derived from the control hierarchy. This makes it possible to utilize hierarchical alarm management.

In our system design, every alarm in the hierarchy has the "enabled/disabled" flag, and the disabled setting in a node overrides the settings in all the children nodes down the hierarchy tree. This choice makes it easy to stop an alarm flood if it starts developing.

The set of alarm recipients specified in a node applies by recursion to all the children nodes down the hierarchy tree. This choice is due to fact that related alarms are likely to be sent to the same auditors and enterprise personnel. To simplify further the management of alarm recipients, we have chosen to follow a role-based approach in our pilot study. More specifically, the recipients of alarms are not individuals, but rather roles such the director of internal audit, or the manager of the divisional IT department.

Finally, the CMBPC system needs built-in logic that monitors the generated alarm time series, decides whether sufficient time has passed before a subsequent alarm has to be generated, and initiates escalation procedures if certain alarms persist for a significant time period. While in our pilot study MCL is not capable of interrupting business operations (since the internal auditors have no operational control), the escalation procedures can be extremely severe, involving in the extreme cases the notification of the CEO of the corporation. These important characteristics of effectively managing alarms are not always considered by CA software developers, thereby allowing the concerns cited above to further impede the rapid adoption of CA in the market place.

7. Audit Trail of CMBPC

Any CMBPC system has to retain sufficient information to provide evidence that the necessary audit procedures were indeed carried out, and to justify the actions that were taken or not taken. This documentation requirement can be satisfied by creating in essence an "audit trail" of CMBPC. Various applicable ways of safeguarding this trail are discussed by Alles et al (2004a). The lower time limit on the retention of this audit trail is determined by the existing standards and statutes, while the upper limit is mostly due to technical considerations.

What has to be included in the audit trail of CMBPC? Clearly, it has to include the history of configuration settings of the CMBPC system and the logging of all the user activities in this system (including the identity of users changing the system configuration).

Should the CMBPC audit trail retain any source information? BP control settings are not voluminous compared with the volume of business transactions and can (and should) be retained in the audit trail of MCL. Note that it is sufficient to keep recording only the changes in the control settings. The complete set of setting has to be recorded only periodically (with relatively low frequency).

Finally, the history of exceptions and alarms also has to be retained in the audit trail. Under the assumption that the CMBPC system is configured properly to prevent the alarm flood from happening, the preservation of the complete operation history of MCL is not prohibitively expensive.

8. Developers of Continuous Monitoring Software

Our pilot study experience has shown that it is feasible for a large internal audit shop to implement a vast array of CA-type procedures to mitigate business risks in certain high impact areas, and to achieve labor savings through automation of audit tasks. However, this way of implementing CA is equivalent to deploying a "home-brewed" ERP system. While there are certainly examples of successful ERP implementations which were programmed in-house, the experience with ERP implementations over the last decade seems to suggest that the wide-scale deployment and long-term success of CMBPC implementations will rest on the availability of well-developed versatile packaged solutions.

While there is always a possibility that a start-up company focused on developing and selling a CMBPC solution can succeed in the marketplace, there are significant barriers to entry. It seems more likely that successful CMBPC packages will be created by solution providers in one or more of the existing three categories: enterprise software vendors, large public accounting firms, and established audit software vendors.

Enterprise software vendors traditionally provided very limited continuous monitoring capabilities within their systems. While modern ERP systems do provide some limited useful functionality, see e.g., SAP's Audit Information System, we are not aware of any major developments or any strategic decision by a major ERP vendor to invest in the development of a fully-fledged CMBPC package. Their often quoted reason is the lack of demand. They argue that since assurance does not contribute directly to the bottom line, CA capabilities, while being a nice extra, do not add a strong selling point to their packages.

Large public accounting firms have been experimenting with continuous monitoring software for a while, and have presented some very interesting research developments at professional meetings (e.g., KPMG's KOLA). At the same time, they seem to remain ambivalent about this development and question its value proposition and likely return on investment. What contributes to their ambivalence is the current focus on external auditing. Indeed, Alles, Kogan, and Vasarhelyi (2005) argue the possible incompatibility between the requirements of Section 201 of the Sarbanes-Oxley Act of 2002 and the implementation of CA by external auditors. As a result, if large public accounting firms do invest in the development of packaged CMBPC solutions, they may be able to utilize these packages themselves only in their internal audit practice, which their external audit clients are not allowed to outsource to any more.

Established audit software (CAAT) vendors have domain knowledge and well-developed libraries of audit tests, and see an opportunity to leverage this intellectual property in the emerging field of CA. For example, ACL has been recently promoting their Continuous Controls Monitoring solutions, such as one for the purchase-to-payment cycle. While CAAT vendors do have a very strong background in data-oriented audit procedures, these vendors are newcomers to the area of system controls auditing and CMBPC packages. They will be facing a significant learning curve to overcome.

While the Sarbanes-Oxley Act of 2002 did introduce some significant constrains which may hinder the development of CMBPC packages by large public accounting firms, the act has also created a window of opportunity to sell CMBPC software as a Section 404 compliance tool. Hopefully, this opportunity, together with the direct labor cost savings that internal auditors will derive from CMBPC implementations, will sufficiently stimulate the development of fully-fledged CMBPC packages and their implementation in the near future.

9. Concluding Remarks

In this paper we report on the approach we have developed and the lessons we have learned in an implementation of the monitoring and control layer for continuous monitoring of business process controls in the US internal IT audit department of Siemens Corporation's US operations. The architecture designed and developed by us within a real world audit application implements a completely independent CMBPC system running on top of its own relational database which has read-only interaction with the application tier of the enterprise information system.

Among our key conclusions is that "formalizability" of audit procedures and audit judgment is grossly underestimated. Additionally, while cost savings and expedience force the implementation to closely follow the existing and approved traditional internal audit program, a certain level of reengineering of audit processes is inevitable due to the necessity to separate formalizable and non-formalizable parts of the program.

Our study identifies the management of audit alarms and the prevention of the alarm floods as critical tasks in the CMBPC implementation process. We develop an approach to solving these problems utilizing the hierarchical structure of alarms and the role-based approach to assigning alarm destinations. We also discuss the content of the audit trail of CMBPC.

Our final conclusion from our pilot study is that the technology needed to implement CMBPC is already available, the laws and standards are (mostly) in place, and the time for initial wide-scale implementations is now. Only diverse practical experience will provide the facts necessary for identifying trade-offs between effectiveness, efficiency and timeliness of audit procedures and determining how to make CMBPC implementations worthwhile.

Acknowledgments

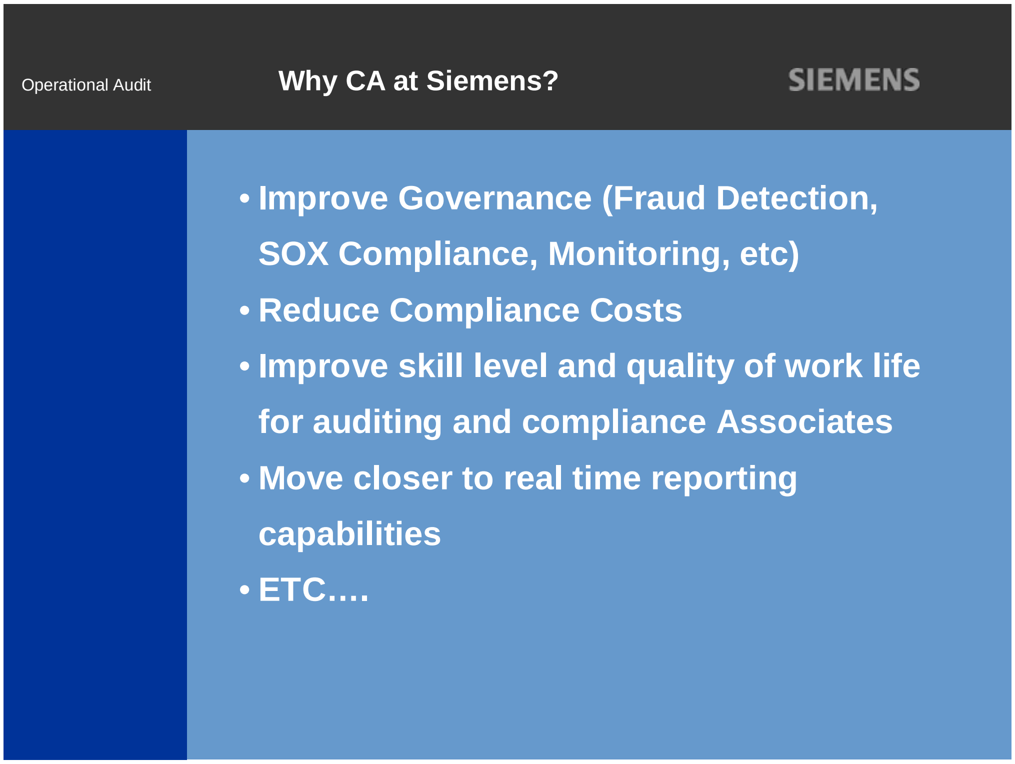
We want to express our thanks to our PhD students Yu Cong, Vasundhara Chaudhuri and Udi Hoitash who worked on implementing this pilot study.

References

1. Alles, M.A., Kogan, A., and Vasarhelyi, M.A. 2002. Feasibility and Economics of Continuous Assurance. *Auditing: A Journal of Practice and Theory*, Vol. 21, No. 1, 125 – 138 (March).
2. Alles, M.A., Kogan, A., and Vasarhelyi, M.A. 2004. Real Time Reporting and Assurance: Have Their Time Come? *ICFAI Reader*, Special issue: Finance in 2004, Institute of Chartered Financial Analysts of India.
3. Alles, M.A., Kogan, A., and Vasarhelyi, M.A. 2004a, Restoring Auditor Credibility: Tertiary Monitoring and Logging of Continuous Assurance Systems. *International Journal of Accounting Information Systems*, Vol. 5, No. 2, 183-202 (July).

4. Alles, M.A., Kogan, A., and Vasarhelyi, M.A. 2005. Implications of Section 201 of the Sarbanes-Oxley Act: The Role of the Audit Committee in Managing the Informational Costs of the Restriction on Auditors Engaging in Consulting. *International Journal of Disclosure and Governance*, Vol. 2, No. 1, 9-26 (February).
5. CICA/AICPA. 1999. Continuous Auditing. Research Report, Toronto, Canada: The Canadian Institute of Chartered Accountants.
6. Davenport, T.H. and Short, J.E. (1990 Summer). The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review*, 11 – 27.
7. Groomer, S. M. and U. S. Murthy. 1989. Continuous auditing of database applications: An embedded audit module approach. *Journal of Information Systems*, 3 (2): 53-69.
8. Hammer, M. 1990. Reengineering Work: Don't Automate, Obliterate! *Harvard Business Review*. July-August.
9. Kogan, A., E.F. Sudit and M.A. Vasarhelyi. 1999. Continuous Online Auditing: A Program of Research. *Journal of Information Systems*, Vol. 13, No. 2, 87 – 103 (Fall).
10. Rezaee, A., R. Elam, and A. Sharbatoghlie. 2002. Continuous Auditing: Building Automated Auditing Capability. *Auditing: A Journal of Practice and Theory*, Spring.
11. Vasarhelyi, M.A. 2002. Concepts in Continuous Assurance. In S. Sutton and V. Arnold, *Researching Accounting as an Information Discipline*, American Accounting Association, Sarasota, Florida.
12. Vasarhelyi, M.A., Alles, M.A. and Kogan, A. 2004. Principles of Analytic Monitoring for Continuous Assurance. *Journal of Emerging Technologies in Accounting*, Vol. 1, No. 1, 1-21.
13. Vasarhelyi, M.A and M.L. Greenstein. 2003. Underlying Principles of the Electronization of Business: A Research Agenda, *International Journal of Accounting Information Systems*, 49 (2003) pp. 1-25.
14. Vasarhelyi, M.A. and Halper, F. 1991. The Continuous Audit of Online Systems. *Auditing: A Journal of Practice and Theory*, 10 (1) 110-125.

Figure 1: CA motivation and value propositions at Siemens

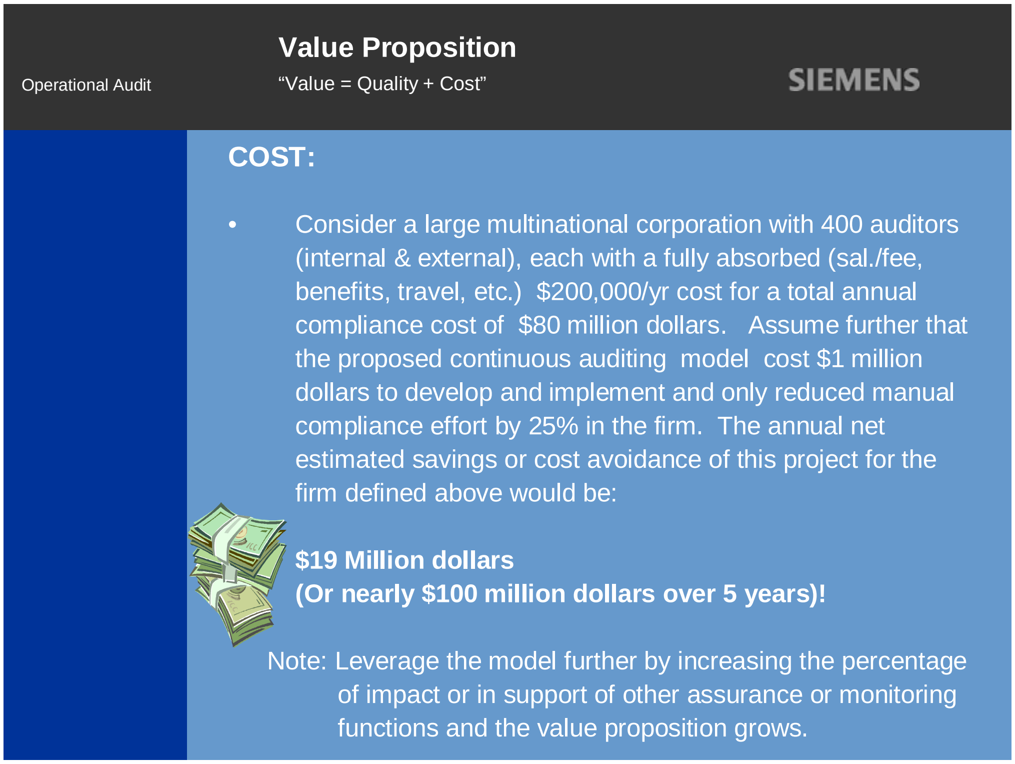


Operational Audit

Why CA at Siemens?

SIEMENS

- Improve Governance (Fraud Detection, SOX Compliance, Monitoring, etc)
- Reduce Compliance Costs
- Improve skill level and quality of work life for auditing and compliance Associates
- Move closer to real time reporting capabilities
- ETC….



Operational Audit

Value Proposition

"Value = Quality + Cost"

SIEMENS

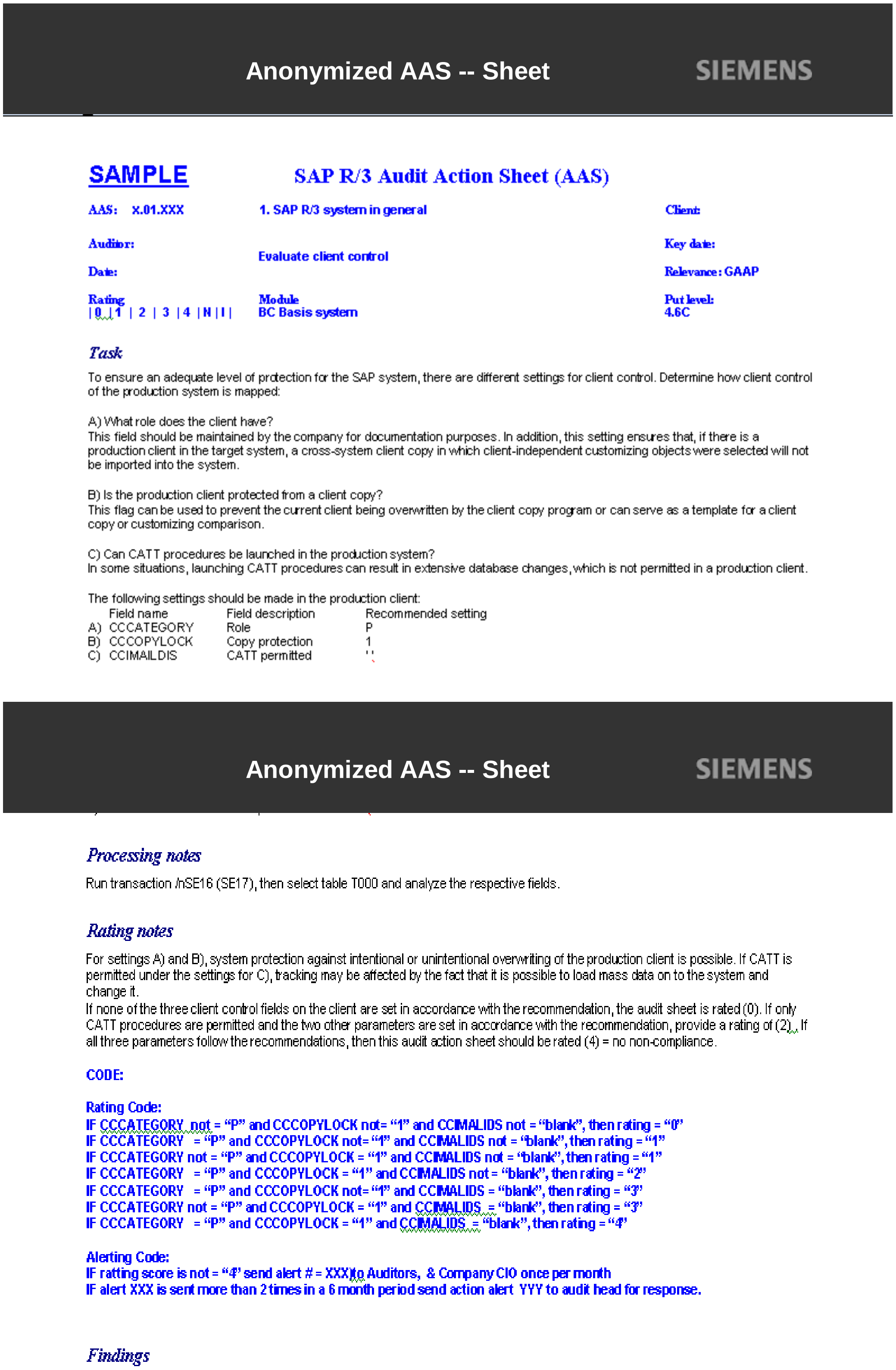
COST:

- Consider a large multinational corporation with 400 auditors (internal & external), each with a fully absorbed (sal./fee, benefits, travel, etc.) $200,000/yr cost for a total annual compliance cost of $80 million dollars. Assume further that the proposed continuous auditing model cost $1 million dollars to develop and implement and only reduced manual compliance effort by 25% in the firm. The annual net estimated savings or cost avoidance of this project for the firm defined above would be:

$19 Million dollars
(Or nearly $100 million dollars over 5 years)!

Note: Leverage the model further by increasing the percentage of impact or in support of other assurance or monitoring functions and the value proposition grows.

Figure 2: Sample Audit Action Sheet with pseudo code version



Anonymized AAS -- Sheet SIEMENS

SAMPLE **SAP R/3 Audit Action Sheet (AAS)**

AAS: X.01.XXX **1. SAP R/3 system in general** **Client:**

Auditor: **Evaluate client control** **Key date:**

Date: **Relevance: GAAP**

Rating
| 0 | 1 | 2 | 3 | 4 | N | I |

Module
BC Basis system

Put level:
4.6C

Task

To ensure an adequate level of protection for the SAP system, there are different settings for client control. Determine how client control of the production system is mapped:

A) What role does the client have?
This field should be maintained by the company for documentation purposes. In addition, this setting ensures that, if there is a production client in the target system, a cross-system client copy in which client-independent customizing objects were selected will not be imported into the system.

B) Is the production client protected from a client copy?
This flag can be used to prevent the current client being overwritten by the client copy program or can serve as a template for a client copy or customizing comparison.

C) Can CATT procedures be launched in the production system?
In some situations, launching CATT procedures can result in extensive database changes, which is not permitted in a production client.

The following settings should be made in the production client:

	Field name	Field description	Recommended setting
A)	CCCATEGORY	Role	P
B)	CCCOPYLOCK	Copy protection	1
C)	CCIMAILDIS	CATT permitted	' '

Anonymized AAS -- Sheet SIEMENS

Processing notes

Run transaction /nSE16 (SE17), then select table T000 and analyze the respective fields.

Rating notes

For settings A) and B), system protection against intentional or unintentional overwriting of the production client is possible. If CATT is permitted under the settings for C), tracking may be affected by the fact that it is possible to load mass data on to the system and change it.
If none of the three client control fields on the client are set in accordance with the recommendation, the audit sheet is rated (0). If only CATT procedures are permitted and the two other parameters are set in accordance with the recommendation, provide a rating of (2). If all three parameters follow the recommendations, then this audit action sheet should be rated (4) = no non-compliance.

CODE:

```
Rating Code:
IF CCCATEGORY  not = “P” and CCCOPYLOCK not= “1” and CCIMALIDS not = “blank”, then rating = “0”
IF CCCATEGORY   = “P” and CCCOPYLOCK not= “1” and CCIMALIDS not = “blank”, then rating = “1”
IF CCCATEGORY not = “P” and CCCOPYLOCK = “1” and CCIMALIDS not = “blank”, then rating = “1”
IF CCCATEGORY   = “P” and CCCOPYLOCK = “1” and CCIMALIDS not = “blank”, then rating = “2”
IF CCCATEGORY   = “P” and CCCOPYLOCK not= “1” and CCIMALIDS = “blank”, then rating = “3”
IF CCCATEGORY not = “P” and CCCOPYLOCK = “1” and CCIMALIDS  = “blank”, then rating = “3”
IF CCCATEGORY   = “P” and CCCOPYLOCK = “1” and CCIMALIDS  = “blank”, then rating = “4”

Alerting Code:
IF ratting score is not = “4” send alert # = XXX to Auditors,  & Company CIO once per month
IF alert XXX is sent more than 2 times in a 6 month period send action alert  YYY to audit head for response.
```

Findings

Figure 3: Architecture of the generic CMBPC System

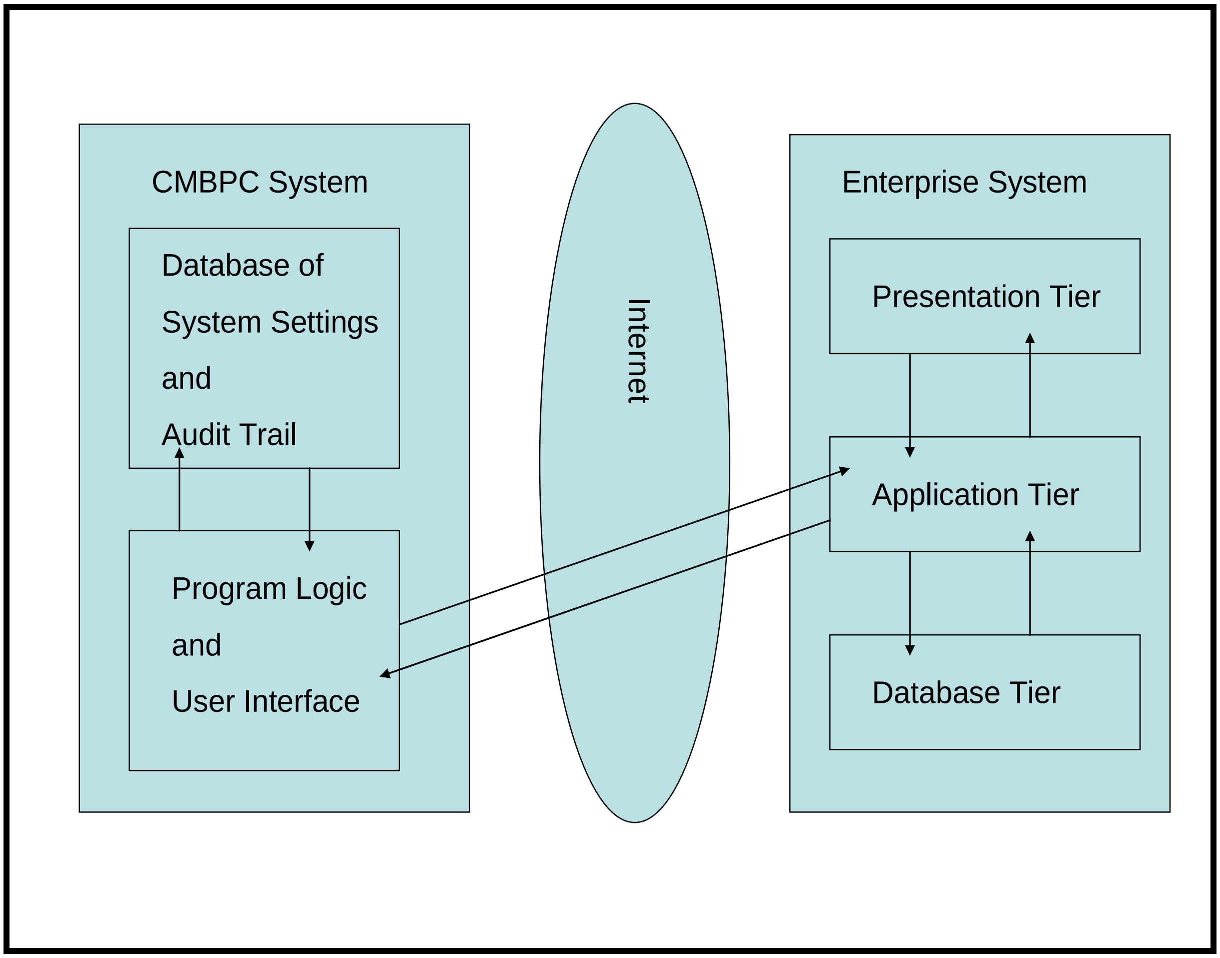
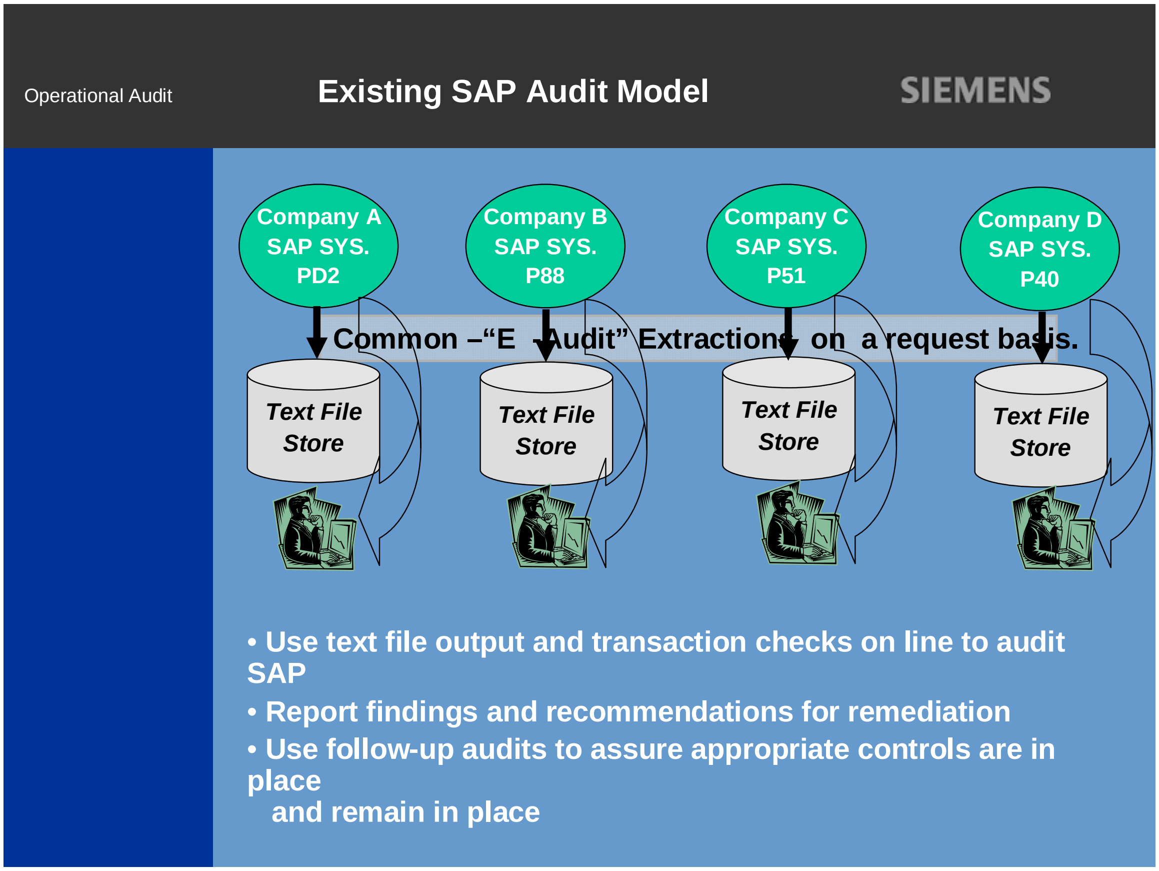


Figure 4: Existing audit procedure with E-Audit data extraction



Operational Audit

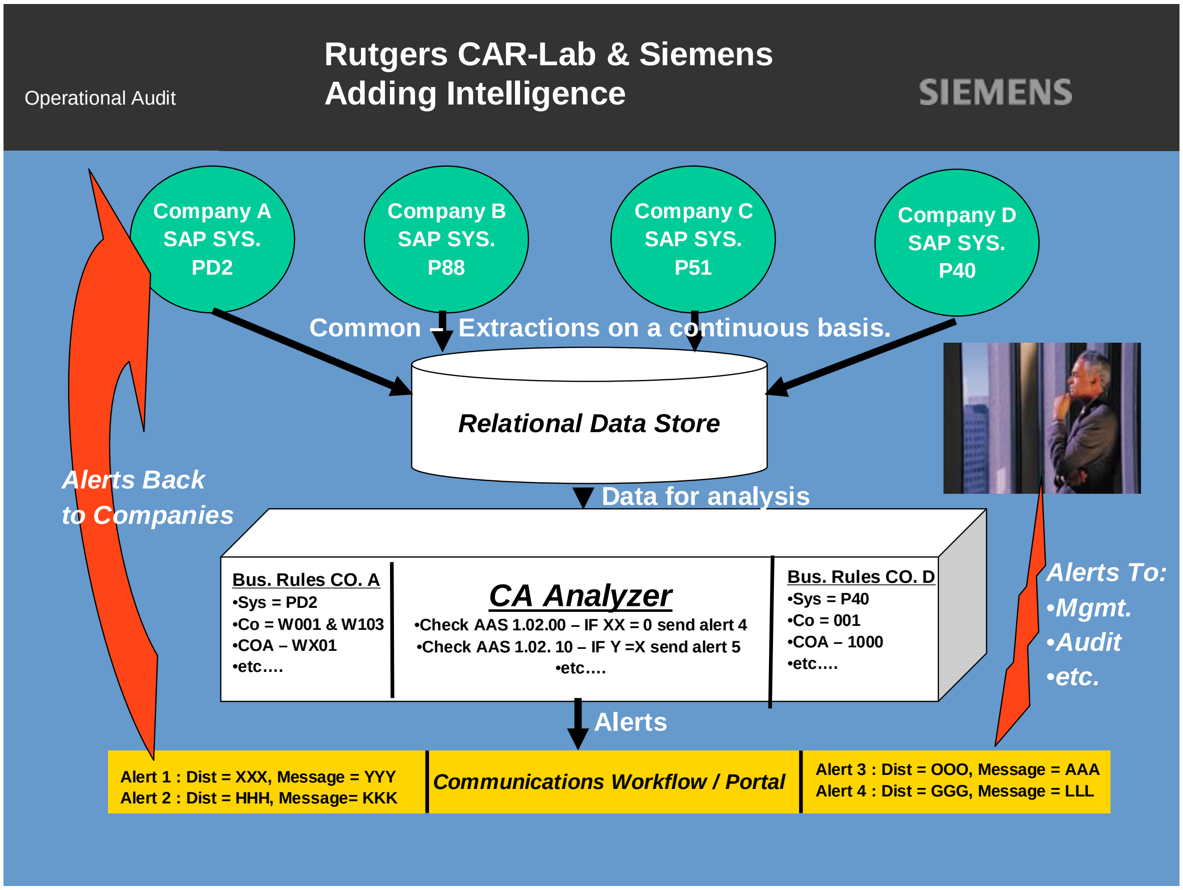
Existing SAP Audit Model

SIEMENS

Common –“E -Audit” Extractions on a request basis.

- Use text file output and transaction checks on line to audit SAP
- Report findings and recommendations for remediation
- Use follow-up audits to assure appropriate controls are in place
  and remain in place

Figure 5: CA-enabled audit procedure



Operational Audit

Rutgers CAR-Lab & Siemens
Adding Intelligence

SIEMENS

Company A
SAP SYS.
PD2

Company B
SAP SYS.
P88

Company C
SAP SYS.
P51

Company D
SAP SYS.
P40

Common – Extractions on a continuous basis.

Relational Data Store

Data for analysis

Alerts Back to Companies

Bus. Rules CO. A
- Sys = PD2
- Co = W001 & W103
- COA – WX01
- etc….

CA Analyzer
- Check AAS 1.02.00 – IF XX = 0 send alert 4
- Check AAS 1.02. 10 – IF Y =X send alert 5
- etc….

Bus. Rules CO. D
- Sys = P40
- Co = 001
- COA – 1000
- etc….

Alerts To:
- *Mgmt.*
- *Audit*
- *etc.*

Alerts

Alert 1 : Dist = XXX, Message = YYY
Alert 2 : Dist = HHH, Message= KKK

Communications Workflow / Portal

Alert 3 : Dist = OOO, Message = AAA
Alert 4 : Dist = GGG, Message = LLL