

Effect of Using Automated Auditing Tools on Detecting Compliance Failures in Unmanaged Processes

Yurdaer Doganata and Francisco Curbera

IBM T J Watson Research Center, 19 Skyline Drive, Hawthorne NY 10532
{yurdaer, curbera}@us.ibm.com

Abstract. The effect of using automated auditing tools to detect compliance failures in unmanaged business processes is investigated. In the absence of a process execution engine, compliance of an unmanaged business process is tracked by using an auditing tool developed based on business provenance technology or employing auditors. Since budget constraints limit employing auditors to evaluate all process instances, a methodology is devised to use both expert opinion on a limited set of process instances and the results produced by fallible automated audit machines on all process instances. An improvement factor is defined based on the average number of non-compliant process instances detected and it is shown that the improvement depends on the prevalence of non-compliance in the process as well as the sensitivity and the specificity of the audit machine.

Topics covered: BPM Governance and Compliance; Management Issues and Empirical Studies; Non-traditional BPM Scenarios.

1 Introduction

The operations of many businesses depend on business processes that rely heavily on human interactions, supported by collaboration software such as e-mail, calendar systems, and others. These processes are highly unstructured, often lack proper documentation and require human intervention as part of the process. The transitions between such unmanaged process activities are not always automated by software components; hence they cannot be fully controlled and monitored by utilizing process execution engines. In the absence of process automation software that can control and record resource and organizational access (who did what and when), compliance check is a costly and time consuming task performed manually by auditors.

Business provenance technology is proposed in [1] to increase the traceability of such unmanaged or partially managed processes. This technology provides for a generic data model, a middleware infrastructure to collect and correlate business events and a query interface to inspect which tasks are executed, when and what resources are involved. Information is selectively captured together with the context of in which it is used, and is then applied to detect compliance violations, in an interactive or automated way. Business provenance technology enables building automated

auditing systems and tools to detect compliance failures continuously and reduce the cost of employing auditors significantly.

Automated continuous auditing systems provide for an almost cost-free auditing opportunity if the initial cost of building such a system is excluded. Such a system can run continuously and performs evaluation for all process instances without adding to the cost of auditing. While continuous audit systems eliminate or reduce the dependency on audit professionals, they are not infallible. The tools that are built to realize automated continuous auditing rely on information extraction from process events and information, including e-mail transactions between the people within the organizations. The extracted information about the processes may contain errors and due to these errors the decision on the compliance may be faulty. Moreover, the testing of a compliance condition may require a level of text analysis that is not yet available in automated systems. Hence, the automated systems can perform fast and extensive auditing of the internal control points at the cost of making mistakes. As a result, some compliance failures may be missed while some other cases that are compliant may be declared non-compliant.

There are many obvious reasons for organizations to worry about compliance in general. A business that has not taken adequate steps to achieve compliance may of course be subject to serious financial penalty as well as civil and penal consequences. Still, compliance has broader and practical impacts. On a more practical level, compliance ensures the quality of products and services and helps the organizations better control their operations and remain competitive. In short, the impact of non-compliance can be profound.

The cost of improving the status of compliance and reduce the risk of being non-compliant could run into millions of dollars for many organizations [2]. Auditing is a central component of compliance operations. Manual auditing involves the use of subject matter experts, but typically covers only a small set process instances because of time and cost constraints. Audits are performed in a quarterly or yearly basis, and cases are selected through statistical sampling. There is thus a trade of between the cost of sampling sufficient number of cases and the possibility of poor auditing which may cause missing opportunities for corrective action. While traditional audits are performed a few times a year, it is widely believed that compliance is an ongoing process that goes beyond testing and evaluating the internal controls of a sampled space. Thus many corporations focus on enhancing or implementing systems to ensure compliance on a continuous basis [3]. AMR Research survey reveals that the spending of companies on governance, risk management and compliance will increase 7.4% in 2008 and exceed \$32B [4]. As a result, companies invest on implementing automated continuous audit systems [5] that would reduce the cost of compliance and would not be limited to selected instances of business processes due to budget constraints.

In this article, we introduce and measure the effectiveness of an automated continuous audit tool that is designed to detect compliance failures. We measure the effectiveness of the tool by its capacity to detect compliance failures during the execution of an unmanaged business process. This is accomplished by identifying a set internal control points and compare the number of non-compliance instances detected in the presence and in the absence of auditing tool. As a result of this comparison, we quantify how much the traditional auditing process performed by auditors under a budget constraint

can be improved by employing auditing tool. Our approach is based on inferring the prevalence of non-compliance and the performance of the tool from a set of sample test results. We then use the inferred results to calculate the improvement as detailed in sections 5-7.

Next section briefly overviews the business provenance technology that we employed to implement the automated auditing tool. In Section 3, an e-mail based business process is described which is used to evaluate the effectiveness of the tool. This process is selected as a typical human centric process where an execution engine is not used to control and manage the process. Hence, traditionally auditing is done by employing subject matter experts. A set of internal key control points are defined to determine the status of compliance and as a basis for our comparative study. In Section 4, a mathematical model is presented for faulty auditing tools for which the statistical performance measures are inferred in section 5. A methodology is proposed to measure the effectiveness of the automated machines in Section 6 and the numerical results are presented in Section 7. We conclude in Section 8.

2 Related Work

Key control points within the business process help identifying risks throughout the organization before they cause integrity lapses. A risk classification associated with BPM cycle provided in [6] and mentions CobIT framework as a set of audit-oriented guidelines create control objectives aligned with the BPM life cycle concept. The key control points that we used to measure the effectiveness of automated audit tools are driven from the rules and regulations in business documents and specifications written in natural language.

A formal representation of these key control points are not within the scope of this paper, but we briefly mentioned the methodology we adopted below. Regardless, there has been a number of works in the literature focusing on formal representation of internal control points by using various rule languages. A compliance metamodel for formally capturing key control points and managing them in systematic lifecycle is presented in [7]. A formal system for business contract representation with reasoning about violations of obligations in the contacts is proposed in [10]. Various aspects of Business Contract Language (BCL) [9] are evaluated by using a logic-based formalism called Formal Contract Language (FCL) in [11] and the need to ensure compatibility between business processes and business contracts is addressed. In [13], a rule language, RuleML, is proposed to express business rules explicitly as a better alternative to other XML languages such as BCL and XrML [15]. In business provenance technology [1] that is employed in this paper, key control points are directly expressed in terms of the business provenance entities which are the nodes and edges of the provenance graph formed during the execution of business operations. While our approach does not require logical analysis of the business rules and is implemented by employing a simple SQL/XPATH based query interface to the provenance store, the rule developers are expected to aware of the runtime operational environment. Hence, it lacks the reusability and the flexibility of formal representation systems.

Similarly [12] proposes a framework to ensure semantic correctness of the process when ad-hoc process instance deviations occur or when modeling process templates. The domain knowledge is integrated into process management system as process constraints and each constraint is expressed in terms of source and target tasks, their orders and user defined parameters. The expressiveness of the presented constraints is, however, limited to task level, data is not included. We use the attributes of all business provenance data (data, task, resource, process) and their relation to express key control points.

A number of works [8], [14] advocate addressing control objectives early in design time and propose supporting mechanism for business process designers. A method is proposed in [8] to help the process designers to measure the compliance degree of a given process model against the set of objection. A language is introduced in [14] to express temporal rules about the obligations and permissions. In order to help the designers at the process modeling time to validate and verify business contracts. In measuring the effectiveness of compliance tools, we do not assume that the control objectives were known at the time of process design. But the audit tools are designed based on the control objectives over existing business operations.

The method presented in this paper to measure the effectiveness of automated audit tools does not depend on a particular process tracking technology or control point representation. The methodology could be employed to cases that use other formal representations of control points and process tracking technologies.

3 Automated Auditing Tool

An automated auditing tool is a software system that captures information relevant to the internal control points of a business process, puts them into context and computes the compliance status for each control point. Auditing tools rely on correlating the data extracted from the underlying IT system to the relevant aspects of business control points effectively. Hence, relating the business goals to IT level data constitutes the core of this technology as described in [1]. Figure 1 outlines the step of building such a system which starts with converting business rules and regulations into compliance goals (Step ①). Compliance goals are identified by examining the business rules and deciding what action steps are needed. In other words, from the business rules expressed in the language of business people, compliance goals are identified (Step ②). This lays the ground work for setting up IT rules for compliance. Once the compliance goals are identified; tasks, activities, resources, artifacts and their relations that are relevant to the identified goal are determined and mapped onto a data model (Step ③). Recording probes collect business artifacts from the underlying information system and maps them onto provenance data (Step ④ and ⑥). A “provenance graph” is then formed with the data objects constituting the nodes and the relations among the data objects the edges. The data objects are correlated by using the compliance goals and the underlying data model (Step ⑤). Business control points are then expressed in terms of data entities extracted from the process execution trace as graph patterns (Step ⑦). Hence, control points provide a bridge between various components of the business operations and the actual data that could be consumed the IT system. A business control point that can be expressed in terms of the

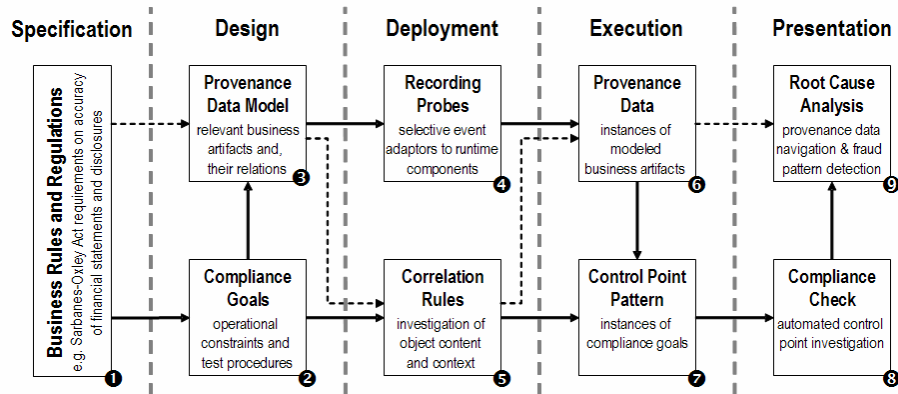


Fig. 1. Steps for compliance checking

data produced and consumed by the IT system can be computed to check compliance in step ⑧. Root cause analysis of compliance failures can be done by querying the provenance graph in step ⑨.

4 Sample e-Mail Based Unmanaged Business Process

In order to measure the effectiveness an automated auditing tool in detecting compliance failures over an unmanaged business process, the following scenario is presented. An e-hosting company manages the customer machines over the internet protected by a firewall and responsible for securing the information assets against unauthorized entry. The company security policy dictates that a firewall manager defines the firewall security controls and ensures on an ongoing basis that firewall policies are implemented using an auditable process. The process is called Firewall Rule Revalidation process and it involves in creating firewall rulesets and communicating these rulesets to the customer and receives approval. The objectives of the process are to ensure both the e-hosting account representatives and the customers understand what rules exist in the customer environment and ensure customer is aware of existing deviations from best practices defined by the e-hosting security policy. If such a process is not implemented, the customer may be at risk due to no longer needed protocols being available for transit traffic, or not being made aware of what protocols are in place and required for support of their environment. The e-hosting company may be held liable for insecure activities, if the customers is not informed of and signs off on the risk involved.

Figure 2 depicts the Firewall Rule Revalidation process where there are three actors of the process, information security advisor, account team and customer.

The responsibilities of these actors are defined as below:

Information Security Advisor (ISA): Prepares firewall rulesets according to the best security practices, modifies them as needed, sends them to the account team and copy to e-mail archive database.

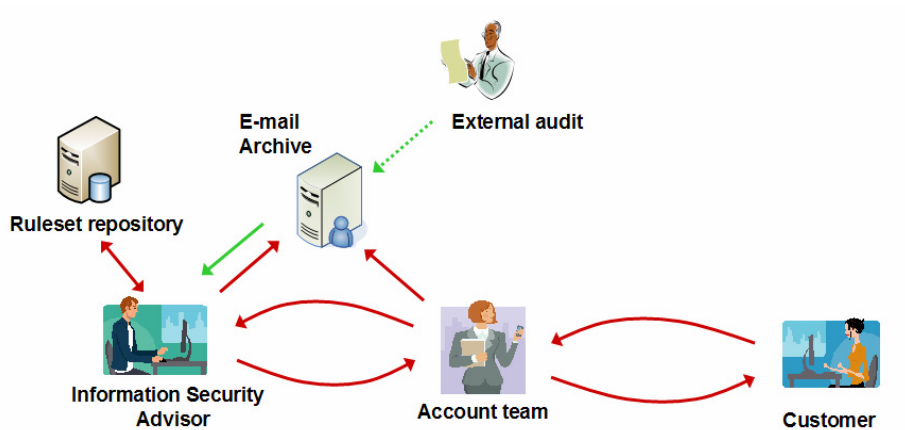


Fig. 2. Firewall Rule Revalidation Process flow

Account Team Member: Receives firewall rulesets from ISA and sends them to customer, records customer response into the e-mail archive.

Customer: Receives firewall rulesets from the account team, reviews them and replies with acceptance or change requests.

Firewall rule revalidation is done once a year. Before the revalidation cycle completes, ISA asks for the firewall rules from the network administrator and checks if the rules are consistent with customer requests and security policies and make modifications if necessary. Once the new rulesets is created, ISA attaches the ruleset to an e-mail and send the e-mail to the account team. Once the ruleset is ready for customer review the process starts. ISA attaches the e-mail to the ruleset and sends to the account team, reminding that it is time for yearly revalidation and ask account team.

4.1 Key-Control Points

In order to assure proper revalidation every year, several internal control points are defined. The compliance of key control points assures that firewall rule revalidation is completed successfully. The description of these internal key control points (KCPs) are described below:

KCP1: A process record exists with a copy of the email from ISA to account team with firewall ruleset is attached.

KCP2: The new ruleset is prepared before the revalidation period ends.

KCP3: A revalidation email must be sent by the account team to the customer within 5 days after email from ISA was received.

KCP4: An acceptance email response from customer must be within 10 days after the first email sent by the account team to the customer.

KCP5: The revalidation process completes within 30 days of being started

KCP6: The revalidation process was completed within the review interval after the prior revalidation completion.

Automated compliance checking process is based on analyzing all the emails in the e-mail archive and classifying them based who sends the e-mail and for what purpose. A data model is built that capture the relevant aspect of the process and according to the description given in [1]. Relationships among data items are extracted by using key control point definitions. As an example, in order to evaluate the status of KCPs, all the e-mail sent by ISAs, Account Team members and customers are examined. Text analysis is used to examine the unstructured parts of the e-mails such as body and subject; the e-mail addresses are extracted from “to” and “from” fields. Based on the extracted relations, each e-mail is scored and labeled as either “from ISA to Account Team” or “from Account Team to Customer” or “from Customer to Account Team”. The relations between the e-mails, their context, receivers and senders are established. The dates of the labeled e-mail are extracted to check the compliance status of each control point.

5 Statistical Modeling Results

The problem of using automated audit machines to determine the compliance failures is equivalent to determining the prevalence of a medical condition through screening the population by using a medical diagnostic test which is not a gold standard [17]. The public health services in many cases use tests which are not 100 percent accurate to estimate the prevalence of the disease. Similarly, the prevalence of non-conformance in a business process can also be estimated by using automated auditing systems which are fallible in making classification for compliance. This is a binary classification problem where the instances of business processes are grouped into two on the basis that they satisfy certain key control points as compliant or not. A practical approach to this classification problem needs to consider two parameters, namely, quality of the classification decisions and cost. For the purpose of the work presented in this paper, we will assume that audits performed by experts always result in a correct classification decision. There is of course a human error factor that we are not factoring in the analysis presented here. On the other hand, there is a considerable cost associated with manual audits, which limits the number of process instances that can be audited this way. The cost of performing an evaluation of compliance by using an automated audit machine can be assumed negligible, allowing in many cases for full coverage of process instances. The results of automated classification, on the other hand are fallible.

In modeling this problem, we define the prevalence as the total number of non-compliant process instances in the population of all process instances. Mathematically, it is the probability that a case is marked as not-compliant by an audit expert, $Pr(I = 1)$. In medical field, this corresponds to the *prevalence*, p , of the disease. Prevalence cannot always be measured by using expert opinion because the cost of employing experts may be prohibitive. However, it is possible to draw inference about the prevalence, p , of non-conformance in a set of execution traces of process instances by using fallible automated auditing tool, if a measure of the auditing system’s performance in identifying non-compliant instances is known. The performance of such an auditing tool is measured by its *sensitivity* and *specificity*. Sensitivity measures the proportions of actual positives (that is non-compliant cases) which are correctly

identified, while specificity measures the proportions of negatives (compliant cases) which are correctly identified. We will refer the probability that a randomly selected instance is actually compliant as $Pr(I = 0)$, the probability that a fallible auditing tool labels an instance compliant as $Pr(F = 0)$ and non-compliant as $Pr(F = 1)$. Hence,

$$\eta, \text{ Sensitivity: } TP / (TP + FN) = Pr(F = 1 / I = 1) \tag{1}$$

$$\theta, \text{ Specificity: } TN / (TN + FP) = Pr(F = 0 / I = 0) \tag{2}$$

where TP is the number of non-compliant instances labeled as non-compliant, FN is the number of non-compliant instances labeled as compliant, TN is the number of compliant instances labeled as compliant, FP is the number of compliant instances labeled as non-compliant. The following joint probabilities of classification $p_{if} = P(I=i, F=f)$, where $i, f = 0, 1$ can be verified easily

$$p_{00} = P(I=0, F=0) = \theta \cdot (1-p) \tag{3}$$

$$p_{10} = P(I=1, F=0) = (1-\eta) \cdot p \tag{4}$$

$$p_{01} = P(I=0, F=1) = (1-\theta) \cdot (1-p) \tag{5}$$

$$p_{11} = P(I=1, F=1) = \eta \cdot p \tag{6}$$

Here p_{if} is the probability that a case is classified as f by an auditing tool when the infallible classifier, i. e. audit expert, determines the case as i where $f, i \in \{0,1\}$.

In this section we will use Bayesian approach to estimate the distributions of η , θ and p based on the test results of the tool on a small set of process instances. Then, we will approximate p_{if} from equations (3) - (6) and compute the effectiveness of the tool by employing these estimations in sections 6 and 7. Following the Bayesian approach in the presence of misclassification, it is a common practice to assume that prior information is in the form of a beta density for prevalence, specificity and sensitivity [17]. The reason for selecting Beta distribution is that it is a flexible family of distribution and a wide variety of density shapes can be derived by changing the associated parameters of the beta distribution [21]. It is also a conjugate prior distribution for the binomial likelihood which simplifies the derivation of the posterior distribution significantly. The probability density function of a beta distribution with parameters (α, β) is given by

$$f(\varphi) = \begin{cases} \frac{1}{B(\alpha-1)} \varphi^{\alpha-1} (1-\varphi)^{\beta-1}, & 0 \leq \varphi \leq 1, \alpha, \beta > 0, \quad \text{and} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

We will hence assume that the prior information on p , η and θ is expressed through independent beta distributions as $Beta(\alpha, \beta)$, $Beta(\alpha_1, \beta_1)$ and $Beta(\alpha_2, \beta_2)$ respectively. Our purpose is to infer posterior distributions of p, η and θ after observing the compliance evaluation results of the auditing tool. Let the auditing tool with sensitivity η and specificity θ evaluates the compliance status of a key control point for N process instances, for which the truth values (actual compliance status) are known. As

a result, let $n_{\bullet 1} = n_{11} + n_{01}$ instances are marked as non-compliant ($F=1$) and $n_{\bullet 0} = n_{10} + n_{00}$ instances are marked as compliant ($F=0$) where n_{11} , n_{10} , n_{01} and n_{00} are the number true positives, false negatives, false positives and true negatives respectively as shown in Table 1.

In the presence of the observed data, the posterior distributions of p , η and θ are still independent Beta distributions [21]. This can be shown directly by using the Bayesian theorem that the posterior joint distribution is the product of the likelihood function of the observed data (*binomial*) and the prior distribution (*beta*). Hence, the joint posterior distributions of p , η and θ are found as

$$p \sim \text{Beta}(\alpha + N_1, \beta + N_0) \tag{8}$$

$$\eta \sim \text{Beta}(\alpha_1 + n_{11}, \beta_1 + n_{10}) \tag{9}$$

$$\theta \sim \text{Beta}(\alpha_2 + n_{00}, \beta_2 + n_{01}) \tag{10}$$

Equations (9) and (10) are the posterior distributions of the sensitivity and the specificity of the auditing tool that produced n_{11} true positives, n_{10} false negatives, n_{11} , true positives and n_{01} false positives in a business environment where the prevalence of non-compliance is p . From these observed test results, inference about the marginal distributions for sensitivity and specificity is possible by using Gibbs sampler algorithm [23] as will be shown next.

6 Inference of Marginal Densities p , η , θ .

Inference about prevalence $p \sim \text{Beta}(\alpha, \beta)$, sensitivity $\eta \sim \text{Beta}(\alpha_1, \beta_1)$ and specificity $\theta \sim \text{Beta}(\alpha_2, \beta_2)$ can be drawn by running a test using the auditing tool and observing true positives and false negatives as depicted in Table 1. The technique is well known in the literature as Gibbs sampler algorithm [18]-[20], [23]. Gibbs sampler is an iterative Markov-chain Monte Carlo technique developed to approximate

Table 1. Test results of N sample process instances

		Truth (Audit expert)		
		+	-	
Test (Automated audit)	+	n_{11}	n_{01}	$n_{\bullet 1}$
	-	n_{10}	n_{00}	$n_{\bullet 0}$
		N_1	N_0	N

intractable posterior distributions. The algorithm uses the observed data to compute the posterior distributions of prevalence, specificity and sensitivity by applying Bayes’ theorem and conversely computes the distributions of the observed data by using the prior distributions of prevalence, sensitivity and specificity as described in [23]. Gibbs sampler derives posterior probability distributions that best fit given prior distributions $Beta(\alpha, \beta)$, $Beta(\alpha_1, \beta_1)$ and $Beta(\alpha_2, \beta_2)$ and observed data, n_{11} , n_{10} , n_{11} and n_{01} . As described in [17], arbitrary starting values can be chosen for each parameter. If no prior knowledge or data is available for the initial distributions, α and β parameters are selected as 1 which corresponds to uniform distribution. Gibbs sampler converges to the true values of the posterior distributions after running tens of thousands of iterations.

Table 2. Input values for the prevalence calculator given in [22]

Input values for the prevalence calculator	
Test Results:	
Number of samples tested	N
Number of samples positive:	$n_{11} + n_{01}$
Alpha and Beta parameters for prior distributions:	
Prior prevalence, alpha	N_j
Prior prevalence, beta	$N - N_j$
Prior sensitivity, alpha	n_{11}
Prior sensitivity, beta	n_{10}
Prior specificity, alpha	n_{00}
Prior specificity, beta =	n_{01}
Simulation details:	
Number of iterations:	50K+
Number to discard: 5	5K+
Starting Values	
Number of true positives	n_{11}
Number of false negatives	n_{10}

Reference [22] provides for an on-line calculator to estimate the true prevalence based on testing of individual samples using a test with imperfect sensitivity and/ or specificity. The input values required for the calculator are listed in Table 2. These include the number of samples (process instances) tested, the number of samples labeled positive (non-compliant), α and β parameters for prior prevalence, sensitivity and specificity distributions, number of iterations to be simulated in the Gibbs sampler, number of iterations to be discarded to allow convergence of the model and

initial number of true positives n_{11} and false negatives n_{10} . Table 2 is generated by using the notation given in Table 1. The initial α and β values for prior distributions are selected by using the fact that $\alpha / (\alpha + \beta)$ is the mean value of a beta distribution and by using the observed data given in Table 1. As an example, the mean value of the prior prevalence can be approximated as $N_1 / (N_1 + N_0)$. Hence, the beta value for prior prevalence is approximated as N_0 and the alpha value is approximated as N_1 .

7 Numerical Results

We run our auditing tool over 135 instances of the sample e-mail based firewall rule revalidation process to verify the compliance of the six key control points defined in section 3. 1. By using the prevalence calculator described above and the number of observed true positives, true negatives, false positives and false negatives for each key control point, we inferred the distributions for p , η and θ . The test results of the auditing tool and the associated mean values for p , η and θ are displayed for each key control point in Table 3.

Table 3. Average prevalence, sensitivity and specificity values for the auditing tool inferred for the six key control points

	KCP1	KCP2	KCP3	KCP4	KCP5	KCP6
n_{11} (TP)	7	64	55	41	42	40
n_{01} (FN)	1	5	5	12	5	9
n_{10} (FP)	8	20	31	41	18	6
n_{00} (TN)	119	46	44	41	70	80
$E(p)$	0.074	0.502	0.460	0.399	0.366	0.362
$E(\eta)$	0.826	0.924	0.847	0.759	0.790	0.793
$E(\theta)$	0.934	0.892	0.555	0.496	0.758	0.910

Table 3 implies that the performance of the auditing tool varies for each key control point. This is expected since the data used to compute the compliance of each control point is different. As a result, the effectiveness of the tool also varies for each control point. The inferred prevalence of non-compliance for different key control point shows that the rate of non-compliance is highest for KCP2 ($E(p)=0.502$) and lowest for KCP1 ($E(p) = 0.074$). This information can be used to identify the problematic points in the process.

Once we measure the performance of the auditing tool with its sensitivity and specificity values, we would like to understand how effectively we can use the tool to increase the rate of detecting non-compliant processes. In the next section, we

will propose a method to measure the effectiveness of the automated auditing tool with given sensitivity and specificity in an environment where the prevalence of non-compliance is p .

8 Measuring the Effectiveness of Auditing Tool

Given a fallible auditing tool with sensitivity and specificity (η, θ) , and given a fixed budget to fund the use of audit experts, we would like to find out how much we can improve the detection of non-compliant process instances. As discussed before, poor auditing may cause missing opportunities for corrective action. Hence, we would like to maximize the number of non-compliant cases detected as a result of auditing. On one hand we have a budget constraint which limits the number of cases we can audit by using an expert. On the other hand, we have a fallible automated audit machine which can be used to evaluate every process instance without incurring extra cost. The goal is to devise a methodology for enabling to detect the largest number of non-compliant instances possible under these constraints. One possible methodology is to evaluate all process instances by using the automated audit machine and ask experts randomly re-evaluate M_1 cases among the ones marked as non-compliant (Region N) and M_2 among the ones marked as compliant (Region C) by the automated audit machine. This way the sample space that the experts operate is reduced. We assume that the budget permits the expert evaluation of only $M = M_1 + M_2$ cases. The effectiveness of the proposed methodology can be measured by comparing the expected number of non-compliant process instances detected. If the number is higher than what experts would have determined under budget constraint without using the methodology, then we can conclude that the methodology improves the auditing process in general.

The probability that a randomly selected process instance is labeled non-compliant, $P(F=1)$, by the auditing tool is $(p_{11} + p_{01})$. Given the condition that the auditors work only on instances labeled as non-compliant by the tool (Region N), probability that the auditors detect a non-compliant case is $Pr(I=1/F=1) = Pr(I=1, F=1)/P(F=1) = p_{11} / (p_{11} + p_{01})$. Similarly, the probability that an auditors detects a non-compliant cases among the ones labeled as compliant by the tool (Region C) is $Pr(I=1/F=0) = p_{10} / (p_{00} + p_{10})$. Hence, the average number of non-compliant cases detected by using this method can then be found as below where the function W is called the “worth” of this method.

$$W = M_1 \frac{p_{11}}{p_{11} + p_{01}} + M_2 \cdot \frac{p_{10}}{p_{10} + p_{00}} \quad (11)$$

The worth function is maximized by making the experts work either in the region labeled as compliant (Region C) or as non-compliant (Region N) depending on the values of $p_{11} / (p_{11} + p_{01})$ and $p_{01} / (p_{00} + p_{01})$ provided that the budget constraint M is less than the size of both regions. This is a reasonable assumption since the size of process instances in both regions are usually much larger than M . Hence,

$$\max\{W\} = \begin{cases} M \frac{P_{11}}{P_{11} + P_{01}} & \frac{P_{01}}{P_{01} + P_{00}} \leq \frac{P_{11}}{P_{11} + P_{01}} \\ M \frac{P_{10}}{P_{10} + P_{00}} & \frac{P_{10}}{P_{10} + P_{00}} > \frac{P_{11}}{P_{11} + P_{01}} \end{cases} \quad (12)$$

In the absence of auditing tool, we would only rely on the efforts of the audit experts. The average worth of this practice would then be the product of M and the prevalence of non-compliance, p . Let W_0 be the worth of using only experts as auditors, the expected worth is then

$$W_0 = Mp \quad (13)$$

Potential improvement of using auditing tool can then be measured by the ratio of the worth functions $\max\{W\}$ and W_0 . From (12) and (13), the improvement function I is found as:

$$I = \begin{cases} \frac{P_{11}}{(P_{11} + P_{01})p} & \frac{P_{01}}{P_{01} + P_{00}} \leq \frac{P_{11}}{P_{11} + P_{01}} \\ \frac{P_{10}}{(P_{10} + P_{00})p} & \frac{P_{10}}{P_{10} + P_{00}} > \frac{P_{11}}{P_{11} + P_{01}} \end{cases}, \quad (14)$$

9 Numerical Results for Improvement

In order to simplify the calculations, we will approximate the variables $p_{11}, p_{01}, p_{10}, p_{00}$ with their mean values by using equations (3)-(6) and the fact that prevalence, sensitivity and specificity are independent beta distributions as $\text{Beta}(\alpha, \beta)$, $\text{Beta}(\alpha_1, \beta_1)$ and $\text{Beta}(\alpha_2, \beta_2)$ respectively as follows:

$$p_{00} \sim E(\theta.(1-p)) = \frac{\alpha_2\beta}{(\alpha_2 + \beta_2)(\alpha + \beta)}, \quad (15)$$

$$p_{10} \sim E((1-\eta)p) = \frac{\alpha\beta_1}{(\alpha_1 + \beta_1)(\alpha + \beta)}, \quad (16)$$

$$p_{01} \sim E((1-\theta).(1-p)) = \frac{\beta_2\beta}{(\alpha_2 + \beta_2)(\alpha + \beta)}, \quad (17)$$

$$p_{11} \sim E(\eta.p) = \frac{\alpha_1\alpha}{(\alpha_1 + \beta_1)(\alpha + \beta)}. \quad (18)$$

By using the expected sensitivity and specificity values displayed in Table 3 and employing equation (15)-(18), the percentage improvement $(I-I) \times 100$ of using auditing tool for each KCPs is calculated and the results are displayed in Table 4.

Table 4 Improvement obtained by using automated auditing tools for key control points

	KCP1	KCP2	KCP3	KCP4	KCP5	KCP6
$E(p)$	0.074	0.502	0.460	0.399	0.366	0.362
$E(\eta)$	0.826	0.924	0.847	0.759	0.790	0.793
$E(\theta)$	0.934	0.892	0.555	0.496	0.758	0.910
I	6.75	1.49	1.34	1.25	1.79	2.30
%	575	49	34	25	79	130

From equation (14) we can conclude that the improvement depends on the prevalence and the probability that a randomly selected process instance is found non-compliant by the expert within each particular region. Depending on the region (labeled compliant or non-compliant) experts work, this probability is either $p_{11} / (p_{11} + p_{01})$ or $p_{10} / (p_{10} + p_{00})$. It is clear from the equation that there is always going to be improvement as long as the prevalence is less than the probability of detecting a non-compliant process instance by using the tool. This is expected since the rate of detecting non-compliant instances by employing only auditors cannot be greater than p .

In order to understand the effect of sensitivity and specificity on the improvement as a function of prevalence, we approximate the improvement by using equation (15)-(18) as follows:

$$I = \frac{1}{p(1 - \psi) + \psi} \tag{19}$$

where

$$\psi = \begin{cases} \frac{1 - \theta}{\eta} & \text{if } \frac{p_{01}}{p_{01} + p_{00}} \leq \frac{p_{11}}{p_{11} + p_{01}} \\ \frac{\theta}{1 - \eta} & \text{if } \frac{p_{01}}{p_{01} + p_{00}} > \frac{p_{11}}{p_{11} + p_{01}} \end{cases} \tag{20}$$

In Figure 3, the percentage improvement is plotted as a function of ψ for various prevalence values changing between 0.1 and 0.9. The performance of the auditing tool for each key control point is also mapped on the same figure. As seen in the figure, the improvement is significantly greater in case of KCP1. On the other hand, using the tools does not give the same improvement for KCP4. In general, the improvement percentage is significantly higher when both the prevalence and ψ are small and it converges to zero as ψ converges to 1. In order to explain this behavior, without lack of generality, let's focus on the case where experts are asked to examine only process instances labeled as non-compliant (Region N) by the automated machine. Hence, equation (20) becomes $\psi = (1 - \theta) / \eta$ and indicates that as the specificity and the sensitivity increases, the value of ψ decreases. In effect, the improvement percentage increases. This is expected since as the sensitivity of an audit machine improves, the likelihood of detecting non-compliant process instances by using the tool improves as

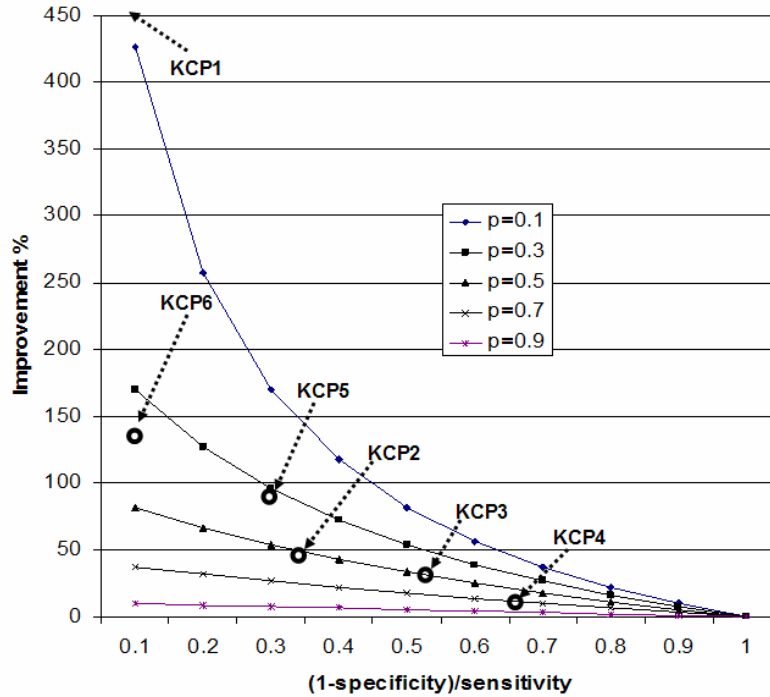


Fig. 3. Percentage of improvement as a function of $(1 - \theta)/\eta$

well. This is why the improvement percentage is higher for smaller ψ values. On the other hand, as ψ approaches to 1, i. e., the sum of the specificity and the sensitivity approaches to one, improvement disappears. The reason for this is that in this case the likelihood of detecting non-compliant process instances approaches to, p , the prevalence. This can be explained as follows: When *sensitivity* is equal to $(1 - specificity)$, the likelihood of having false positives becomes equal to likelihood of having true positives. In other words, the following holds

$$\eta = \frac{TP}{TP + FN} = 1 - \theta = 1 - \frac{TN}{TN + FP} = \frac{FP}{TN + FP}. \tag{21}$$

Equation (21) implies that

$$\frac{TP}{FP} = \frac{TP + FN}{TN + FP}, \tag{22}$$

where TP , TN , FP and FN are the number of true positive, true negative, false positive and false negative observations respectively. Further manipulation of equation (22) yields:

$$\frac{TP}{TP + FP} = \frac{TN + FP}{TN + FP + TP + FN} = p. \quad (24)$$

Here, TP, TN, FP and FN stand for the numbers of true positives, true negatives, false positives and false negatives. Note that this is also equal to the likelihood of detecting non-compliant instances in region N:

$$\frac{p_{11}}{(p_{11} + p_{01})} \approx \frac{TP}{TP + FP} = p \quad \rightarrow \quad I = 1 \quad (25)$$

Equation (25) shows that working in region N does not give any advantage since the detecting a non-compliant process instances in this region is equivalent to the prevalence. The same argument holds for the other region without lack of generality. This means that labeling process instances with the auditing tool does not improve the rate of detecting non-compliant instances if the *sensitivity* of the automated machine is equal to $1 - \textit{specificity}$.

10 Conclusion

The level of compliance of a process, that is, the prevalence of non-compliant instances, can typically be reduced through automation by introducing a business processes management platform and other support middleware such as a content management system. A complementary approach is to increase the levels of compliance monitoring. Processes with low levels of automation, which are essentially unmanaged processes, must rely in an efficient auditing procedure as the only way to prevent systemic non-compliance. Automated auditing tools can be used to complement manual auditing by subject matter experts and expand the amount of process. In this article, we provide a methodology to estimate the effectiveness of these tools. We showed that the effectiveness depends on both the prevalence of non-compliant cases as well as the performance of the tool. The approach is expected to help businesses make smarter decision on employing subject matter experts and utilize automated audit tools.

Our future work will focus on optimizing the use of auditors by taking into account of all operational control points and their correlation.

References

1. Curbera, F., Doganata, Y., Martens, A., Mukhi, M., Slominski, A.: Business Provenance - A Technology to Increase Traceability of End-to-End Operations. In: OTM Conferences vol (1), pp. 100–119 (2008)
2. Greengard, S.: Compliance Software's Bonus Benefits. Business Finance Magazine (February 2004)
3. Gartner.: Simplifying Compliance: Best Practices and Technology, French Caldwell, (Business Process Management Summit (June 6, 2005)

4. Hagerty, J., Hackbush, J., Gaughan, D., Jacobson, S.: The Governance, Risk Management, and Compliance Spending Report, 2008-2009, AMR Research Report, March 25 (2008)
5. Corfield, B.: Managing the cost of compliance, http://justin-taylor.net/webdocs/tip_of_the_iceberg.pdf
6. Zur Muehlen, M., Ho, D.T.: Risk Management in the BPM Lifecycle. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 454–466. Springer, Heidelberg (2006)
7. Christopher, G., Müller, S., Pfitzmann, B.: From Regulatory Policies to Event Monitoring Rules: Towards Model-Driven Compliance Automation. IBM Research Report RZ 3662, IBM Zurich Research Laboratory (2006)
8. Lu, R., Sadiq, S., Governatori, G.: Compliance aware business process design. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) BPM Workshops 2007. LNCS, vol. 4928, pp. 120–131. Springer, Heidelberg (2008)
9. Milosevic, Z., Gibson, S., Linington, J.C., Kulkarni, S.: On Design and implementation of a contract monitoring facility. In: Benatallah, B. (ed.) First IEEE International Workshop on Electronic Contracts, pp. 62–70. IEEE Press, Los Alamitos (2004)
10. Governatori, G., Milosevic, Z.: A Formal Analysis of a Business Contract Language. *International Journal of Cooperative Information Systems* 15(4), 659–685 (2006)
11. Governatori, G., Milosevic, Z., Sadiq, S.: Compliance checking between business processes and business contracts. In: Proceedings of the 10th IEEE Conference on Enterprise Distributed Object Computing (2006)
12. Ly, L.T., Rinderle, S., Dadam, P.: Integration and verification of semantic constraints in adaptive process management systems. *Data and Knowledge Engineering* 64(1), 3–23 (2008)
13. Governatori, G.: Representing Business Contracts in RuleML. *International Journal of Cooperative Information Systems* 14(2–3), 181–216 (2005)
14. Goedertier, S., Vanthienen, J.: Designing compliant business processes with obligations and permissions. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 5–14. Springer, Heidelberg (2006)
15. Lee, J.K., Sohn, M.M.: The eXtensible Rule Markup Language. *Communications of ACM* 46(5), 59–64 (2003)
16. Egizi, C.: High cost of compliance, <http://www.cioupdate.com/career/article.php/3489431/The-High-Cost-of-Compliance.htm>
17. Joseph, L., Gyorkos, T.W., Coupal, L.: Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard. *Am. J. Epidemiol* (1995)
18. Gelfand, A.E., Smith, A.F.M.: Sampling-based approaches to calculating marginal densities. *Journal American Statistics Assoc.* 85, 348–409 (1990)
19. Gelfand, A.E., Hills, S.E., Racine-Poon, A., et al.: Illustration of Bayesian Inference in normal data using Gibbs sampling. *Journal of American Statistics Assoc.* 85, 972–985 (1990)
20. Tanner, M.A.: Tools for statistical inference. Springer, New York (1991)
21. Katsis, A.: Sample size determination of binomial data with the presence of misclassification. *Metrika* 63, 323–329 (2005)
22. Pooled Prevalence Calculator, <http://www.ausvet.com.au/pprev/>
23. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 721–741 (1984)