Model-Based Assurance of Security Controls

Yolanta Beres, Adrian Baldwin, Simon Shiu
Trusted Systems Lab
HP Laboratories Bristol
HPL-2008-7
January 29, 2008*

compliance, assurance, security, audit, metrics

The paper presents an innovative way to assess the effectiveness of security controls where measurable aspects of controls are first captured in the models and then the models are used to analyse the security data gathered from the IT environment. The aim is to lift the risk and security control management lifecycle from a series of people based processes to one where model based technology enhances, connects and where appropriate automates the process. Modelling in such an approach means capturing the relationship between controls and the way the controls should be measured for effectiveness and compliance to regulations and internal policies. This paper also describes how the model based assurance approach has been applied to automate the analysis of critical security controls during several IT application audits. We show advantages both in time savings due to automation of audit testing and in improvement of the control coverage due to the reduction in sampling.

Model-Based Assurance of Security Controls

Yolanta Beres Adrian Baldwin Simon Shiu

HP Labs
Bristol, United Kingdom

yolanta.beres, adrian.badwin,
simon.shiu @hp.com

ABSTRACT

The paper presents an innovative way to assess the effectiveness of security controls where measurable aspects of controls are first captured in the models and then the models are used to analyse the security data gathered from the IT environment. The aim is to lift the risk and security control management lifecycle from a series of people based processes to one where model based technology enhances, connects and where appropriate automates the process. Modelling in such an approach means capturing the relationship between controls and the way the controls should be measured for effectiveness and compliance to regulations and internal policies.

This paper also describes how the model based assurance approach has been applied to automate the analysis of critical security controls during several IT application audits. We show advantages both in time savings due to automation of audit testing and in improvement of the control coverage due to the reduction in sampling.

1. INTRODUCTION

New regulations and constant risk of information-security threats are forcing organizations to more vigorously examine effectiveness of their internal IT controls and processes, including security controls and mechanisms. To deal with these pressures, organizations are calling auditors to make sure their systems comply with corporate security policies and to ensure that appropriate internal controls are implemented to mitigate the security risks to their critical information, applications and infrastructure.

Internal control is broadly defined as a process put into effect by management and other personnel, designed to provide reasonable assurance regarding the achievement of objectives in the following overlapping categories:

- Effectiveness and efficiency of operations;
- Reliability of financial reporting;
- Compliance with applicable laws and regulations.

For a control to be effective, actual results must be compared to expected results or standards, and corrective action must be taken when indicated. The identification of the effectiveness of the mitigating controls usually falls into the responsibility of internal and external corporate auditors. Auditors have developed various methodologies to assess compliance to Sarbanes-Oxley Act (SOX) [1] and other regulations but most of them are still very time consuming and labour intensive. In the world of growing regulatory mandates and industry-based requirements where besides SOX organizations have to meet other regulations such as HIPPA [3] and PCI [2], to name a few, IT management together with auditors are faced with constant pressure to provide more timely and ongoing assurance that controls are working effectively and risk is being managed.

The model based assurance approach presented in this paper aims to model the control framework and use the models to systematically analyse the evidence on the effectiveness of the implemented controls. This immediately lifts the assurance lifecycle from a series of people based processes (risk management, control design and implementation, audit and review) to one where model based technology enhances, connects and where appropriate automates the process.

The paper presents the results of the pilot and investigation into the extent to which the auditing process can be captured in models and automated. The analysis approach allows auditors to dedicate more time to the assessment of risks and the adequacy of controls, rather than manually examining the evidence. It can also allow auditors to deliver timelier and higher-quality results. And, it can help audit management allocate precious — and scarce — staff resources better to focus on high risk or significant areas of exposure to the organization.

The result of the pilot was to provide evidence of the value of the model based approach in a particular and highly relevant context. In addition, by describing the process for creating models, the paper shows the extent to which the proposed modelling approach can transform part of the yearly auditing process into a continuous auditing. Such a transformation goes beyond simplifying the auditing process; it changes the nature of this process, transforming it from data analysis and assessment of deviation into a real-time monitoring and continuous compliance culture.

The paper is organized as follows, section 2 provides a short overview of the architectural components of model based assurance framework. Section 3 describes how models are represented in this framework. Section 4 describes model development process and gives examples of types of analysis and metrics for security controls. Section 5 describes in detail the model that were developed to analyze security controls for

HP's IT application audits. It summarizes the results and benefits, and discusses the implications for continuous compliance. Section 5 discusses related work, with section 6 drawing final conclusions.

2. MODEL-BASED ASSURANCE FRAMEWORK

The main concern of model-based assurance approach is providing assurance that a control, a system, an application or service is 'fit for purpose' and 'well run'. The overall architecture of this approach is shown in figure 1, and was briefly described in [9].

Implementation of the model based assurance approach consists of three phases: (1) model design stage that consists of developing a set of models for a given IT environment (applications or infrastructure) to examine the security processes and controls in place; (2) data collection and extraction stage where the collectors are developed and deployed to get data from the data sources specified in the model; (3) analysis and reporting stage where the previously designed models are used to analyze the collected data and produce a metric-based report comparing the way the controls and system is running against the description in the model. In further sections of this paper we will mainly cover stages (1) and (3). A handful of data collection and extractions frameworks already exist in the market, such as [14], and in our approach we are assuming that information can be made available from a number of sources via a centralized audit store.

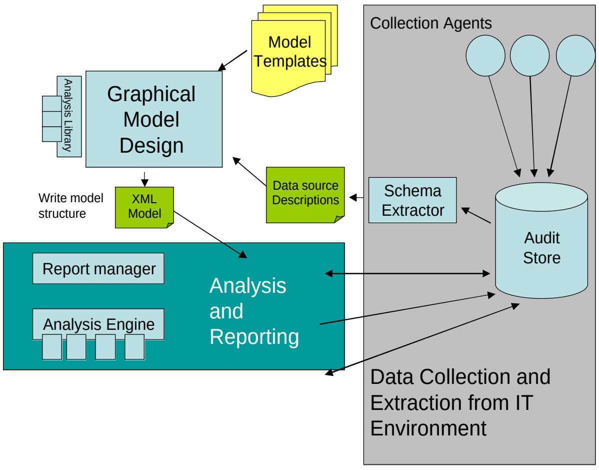


Figure 1. Model-based Assurance Framework.

3. MODEL REPRESENTATION

Our modeling approach supports different types of models. The fundamental one is *the control analysis model* type used for automating the control testing, measurement and the generation of key risk indicators. This type of model works on top of data collected from the IT environment and residing within the audit store. Other model types can be used to pull together the results of the multiple analysis models for comparison purposes or to produce different views of the underlying analysis results for the different stakeholders. These types are called *meta-models*, and can be created in the same way as the control analysis models but instead of linking to the data collected they refer to results from other models either by linking directly to specific model templates or via attributes within the model, such as unique node identifiers. These types of models will not be discussed in detail in this paper, but further explanation can be found in [10].

The control analysis model is represented as a directed acyclic graph structure where the nodes in the graph are analysis nodes that relate to specific metrics or indicators and the edges indicate data flow among the nodes, as seen in the example in figure 2.

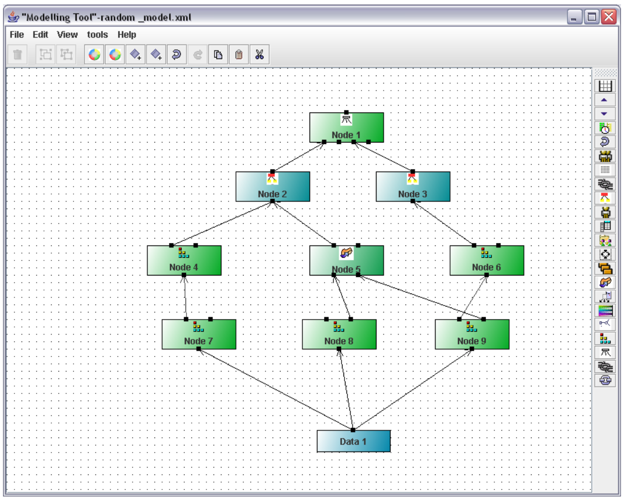


Figure 2. An example model.

An analysis node *an* in the model is defined by 5-tuple <*description*, *type*, *transformation_function*, *input_ports*, *output_ports*>, shown in figure 3, where *description* is used to describe what type of entity the node represents within the control framework, *input_ports* define the data that acts as an input for the transformation function, and *output_ports* define the data produced as an output from the transformation function. The *transformation function* itself is defined by the node's type and is used to derive outputs based on the input data:

$$f_{type} : input\_ports \rightarrow output\_ports \, .$$

The number of the input and output ports that a node has depends on its type. The framework includes a library of analyses nodes types. When building the models a specific analysis type can be selected depending on the metric that has to be produced to indicate the effectiveness of a security control. An array of different analysis node types are available for building the model; for comparing various sets of data, performing checks such as separation of duty, checking event orders etc. For reasoning about overall status of a control, certain types of nodes can be used to define thresholds on specific metrics.

A fully functional model must also consist of one or more data sources. A data source *ds* is also represented as an analysis node, but it only has one *output_port*, with other components empty: <-,-,-,-,*ouput_port*>.

The link between two nodes an_1 and an_2 in a graph is defined by • relation, that specifies the direction of data flow from one node to other and is a transformation function on first node's output ports to produce second node's input ports:

$$f\left(output\_ports_{an_1}\right) = input\_ports_{an_2}\ .$$

The analysis model *AM* can then be formally defined as the transformation relationship between a set of nodes*:*

$AM = \left(AN \times AN, \rightarrow\right)$, where *AN* includes analysis nodes of type *an* and of type *ds*.

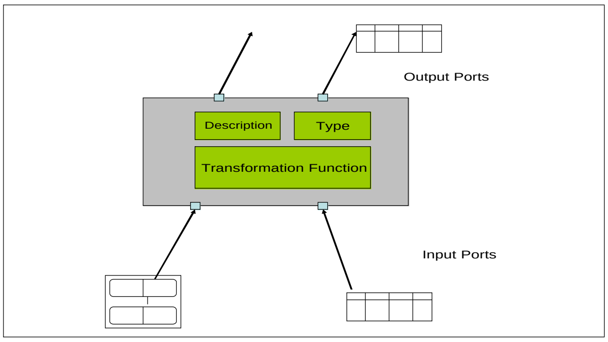


Figure 3. Analysis node representation.

For model creation and visualization we have created a model design tool that allows the user to graphically drawing out the relationships between different parts of the model. New models can be built to relate various controls to high-level policies as well as to detail tests, models can be created by customizing a set of standard templates, for example, the ones that have been already developed to test security controls for compliance to Sarbanes-Oxley requirements.

From the graphical modeling tool the model is rendered into an XML structure that is used as a specification for the analysis and reporting component that creates appropriate data queries on top of the data in the audit store and that generates the metrics and provides views into results that retain the same model structure. Some of the analysis components in the model, mainly at the lower leaf levels, define what raw information has to be gleaned from the IT environment. At the model deployment stage a step then needs to be carried out to determine how best to collect this raw information.

4. MODEL DEVELOPMENT

During the model design stage the aim is to capture in the model the key measurable aspects of a security control that best contribute to evaluating its effectiveness, so minimizing the amount of data that has to be gathered and analyzed. From our pilots we have observed that a key to success is relying heavily on security management processes-related analysis and a limited set of risk indicators rather than just transactional and event-based data. In that respect it is necessary to determine the security controls and IT management processes as well as key indicators that contribute to identifying emerging risk and could be measured on a continuous basis.

It is useful to group the types of measures into two different types:

- Process-based analysis;
- Symptomatic Lagging Indicators.

Process-based analysis and indicators are used to measure an activity or procedure that is part of a control. Such control activities are typically designed by IT management to prevent errors from being introduced into the system, e.g. granting access restrictions to certain capabilities. On the other hand, symptomatic lagging indicators are used to measure the effect of the control activity in the data and detects occurrences of error that may have already been introduced in the system; e.g. a transaction that was improperly authorized. Table 1 below gives examples of the processes and lagging indicators that could be measured to evaluate security-related controls.

Security Control Related Activities	**Process-Based Metrics and Tests**	**Symptomatic Lagging Indicators**
Granting, Modifying and Revoking Access	• Repeatable process for granting and removing access • Privileged system accounts restricted to IT users • Privileges commensurate with job function • Separation of Duty among users • Periodic user reviews • Unique individual Ids	• Total number of users • Shared Accounts • Number of users never logged on • Number of inactive users >60 days • Number of locked users • Number of users with security administration capabilities. • Number of users that can change user master reference data
Password Administration	• Password scanning • Periodic password changes	• Password Settings • Number of default or unchanged passwords
Status and Event Monitoring	• Event monitoring process and follow-up • Security configuration reviews	• Number of security breaches/incidents • Number of unknown system configuration changes

Table 1. Example of indicators associated with security controls.

Ultimately, the aim of the models is to provide views on the status of security controls and to ensure that analysis is performed in a uniform and continuous manner whilst reducing the burden of manual assessment and testing. Once created, the models are deployed to analyze information extracted from the real IT environment assuming that suitable data sources are available.

4.1 Finding data sources

It is often easier to design the model if the structure and meaning of data coming from the IT environment is known in advance. This simplifies selection of the appropriate analysis nodes from the node library. Also the earlier in the project the data sources are identified the better the scope and feasibility of the whole project can be determined, as some control testing may be not easily automatable or certain data sources might need additional collectors to be developed.

The types of data sources required to create the analysis model are identified based on the list of analysis, controls or indicators to be captured in the model. For example, an account approval process analysis will require data about the approvals received and accounts created on the system(s). Similarly, for a patch management process analysis the data is required on patch testing, patch approvals and the information about actual patches applied. Certain information may be available directly through existing monitoring agents, log files or databases. In other cases the IT environment may require additional instrumentation in order to collect the required information.

5. ASSURANCE MODELS AND IT SOX AUDITS

In most organizations the identification of the effectiveness of the mitigating controls usually falls into the responsibility of internal and external auditors, although lately more and more of that responsibility is pushed into IT and security management teams that are ultimately responsible for day-to-day management of IT infrastructure. Due to a lack of a comprehensive set of tools IT auditors usually resort to a manual approach of measuring and testing the controls. This usually involves sampling, downloading data from IT systems and performing manual inspections using spreadsheet-like tools such as audit command language [8]. This approach is labor intensive and time consuming consequentially it can only be done one or twice a year and hence it does not give a continual insight into the effectiveness of controls. The whole approach is labor intensive not only for auditors but also the IT management staff and as such is often viewed as a hindrance rather than good measurement of how security risk is managed.

In piloting our model based approach within the IT audit context, the aim was to examine how much of the manual testing can be lifted and captured in the models, which can then be used to continually analyze and measure the security controls.

Because of new SOX regulation, many of the IT audits have lately been performed in this context. The types of controls that are being audited and measured in this context is usually dictated by following the guidelines from audit governing bodies [4] and by applying best practices as dictated by CoBIT, ITIL, and COSO [5,6,7] framework. Together with maintenance, availability, and continuity controls, security controls are usually evaluated at both application and infrastructure level. This next section describes results of the pilot examining security controls for applications, and in particular access control and authorization controls.

5.1 Application-related security controls

As an input for designing the model and selecting the analysis components we have used controls matrices applied by IT auditors uniformly across all applications. The matrices list set of controls to be tested as part of the audit, together with detailed description of what data has to be gathered and potential measures for evaluating if the control is effective in mitigating perceived risk. In the SOX audit context the security controls tested usually span five separate areas, and are mainly are concerned with access authorizations:

- account termination control
- new user account request process
- correctness of user access privileges
- account usage
- segregation of duty conflicts.

In each control area, based on the testing description and on consultations with the auditors, we then selected a set of analysis nodes and metrics that would best indicate how the control is working. Figure 4 shows the final analysis model with the selected measurements for all five control areas. In the next subsections describe in more detail tests and indicators selected in the areas of account termination and account approval.

We have also created collectors to gather data from several data sources. To get the application data, which in this case were SAP applications, we have created a regular pull of user authorization data from 10 critical SAP applications within HP. In addition, collectors were created to gather data from: (1) Enterprise Directory (ED) on a daily basis about active and terminated employees, (2) account approval email archives on a monthly basis. Besides this real-time data, we were also provided with additional information regarding policies set for some of the controls. For user authorization approval, a list of approvers was provided together with a list of privileges that each approver could approve. For separation of duty, this was a separation of duty matrix showing what privileges are conflicting and should not be assigned to the same user.

5.1.1 Account Termination Controls

Account termination control deals with the mitigation of the risk of existence of active user accounts for terminated and transferred users. It is assumed that this control is working if the following condition is met:

Functional user and system admin accounts are inactivated or deleted within 30 days after the termination or transfer of employees and contractor, or after expiry of account.

To measure effectiveness of this control, auditors currently manually compare the current active user list on the application with the employee list within and Enterprise Directory, a task that sometimes takes a couple of hours, depending on an organization size. Within the model we decided to measure this control with two lagging indicators:

- Number of users with active accounts after employment termination;
- Number of users with expired accounts.

The two indicators are not directly testing the process of account termination in place, but rather the result or effect of this process. If the number of terminated employees with active accounts is high (even one such account might present a risk), this means that the termination control is either not in place or is not working to sufficiently mitigate risks associated with inappropriate user access.

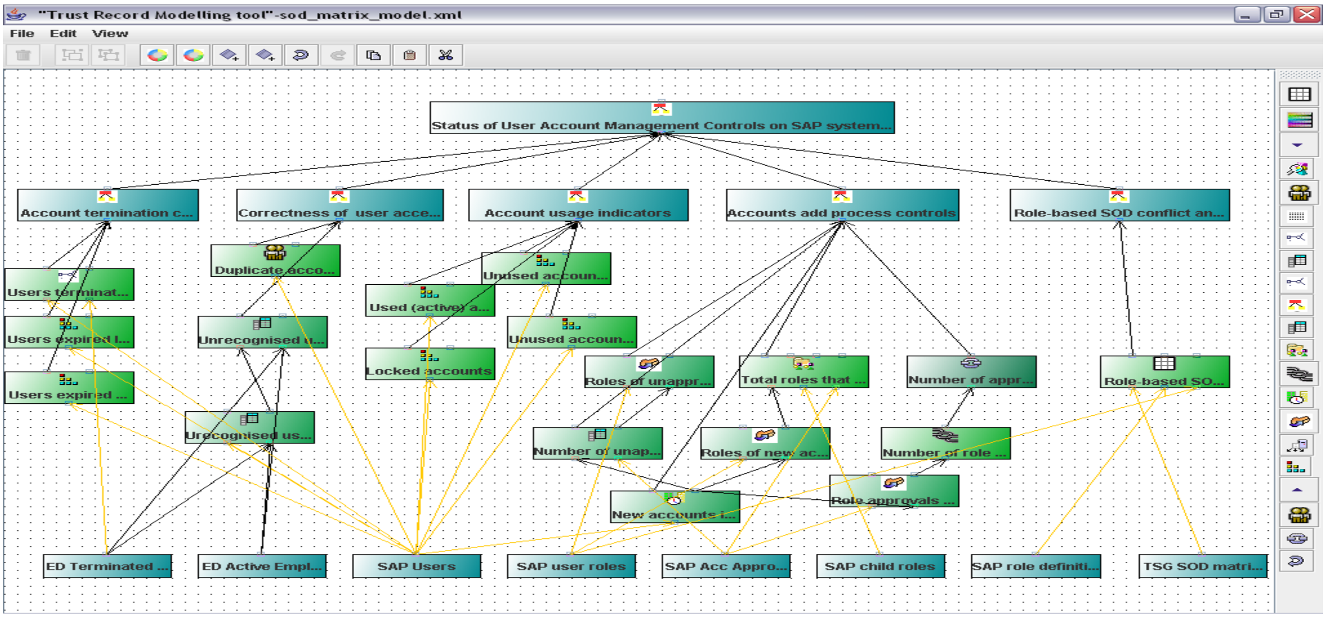


Figure 4. Overall Model for analysis of security controls for an SAP application.

5.1.2 New Access Approval Process

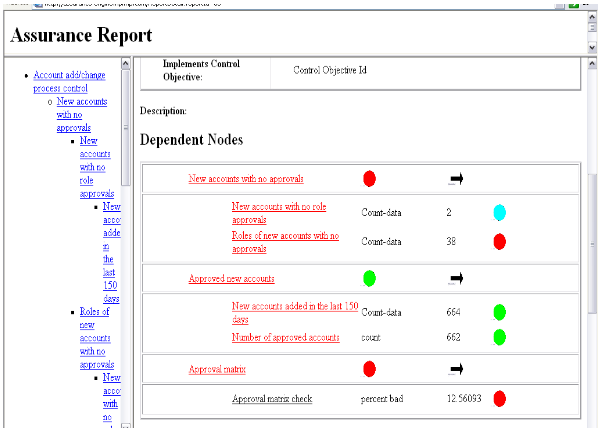
The next important control area is concerned with risk of unauthorized users having access to application data. This requires measuring the account approval process in place so that:

- Request for every new set of privileges (roles) is approved
- Approvals are issued by the appropriate approver as per approval matrix.

This results in a detailed analysis of the approval process (in contrast to the account termination control tests), consisting of metrics that show: (1) the number of the roles that were allocated that were not approved at all, (2) the number of roles that were approved by undocumented approvers.

The current audit practise when testing this particular control is to select a limited number of samples of newly assigned privileges that cover a certain time period, usually from 6 months to a year. For the selected sample privilege allocation the control is then fully tested by locating and examining the associated approvals and authorized approvers list. Sampling is universally accepted as an appropriate testing approach within the audit practice. Full testing would probably be even infeasible with a manual approach[1]. Sampling approach has several limitations, though. Since sample accounts are selected randomly, the final sample might at the end include only accounts for which the control has worked effectively. Such an approach will not necessarily show if during the certain period there wasn't any risks of unauthorised accounts present on the system. It also does not give a full view of how the exceptions were actually handled.

Figure 5 shows top-level results from account approval analysis using model-based approach. Once analysis was run on a periodic basis, auditors were able to view all new account approvals and all account changes. This allowed an auditor to better estimate of how accounts approval process was working overall and if the risk of unauthorised access on the system is properly managed.



Assurance Report

- Account add/change process control
  - New accounts with no approvals
    - New accounts with no role approvals
      - New acco adde in the last 150 days
    - Roles of new accounts with no approvals
      - New acco with no

Implements Control Objective:	Control Objective Id

Description:

Dependent Nodes

New accounts with no approvals	🔴	➡	
New accounts with no role approvals	Count-data	2	●
Roles of new accounts with no approvals	Count-data	38	🔴
Approved new accounts	🟢	➡	
New accounts added in the last 150 days	Count-data	664	🟢
Number of approved accounts	count	662	🟢
Approval matrix	🔴	➡	
Approval matrix check	percent bad	12.56093	🔴

Figure 5. Top-level results from access approval analysis.

5.2 Results

During the application audit pilots the full model that is shown in figure 4 was deployed on a periodic basis (monthly, but sometimes weekly) across 10 critical SAP applications, with results from analysis being used directly to assist in four audit engagements. The benefits of this approach were recognized by both auditors and the application owners.

For auditors this approach saves a lot of time and effort because it minimizes the amount of manual tests they have to do; in

[1] For an application that has 2000-3000 users there are around 600 new accounts per half year with an individual account having 20 or more roles, resulting in 2000-3000 role approvals/denies.

some cases more than 50%[2]. Because the results are available continuously (or the analysis can be performed just before an audit), it also allowed for auditors to see where the problems are and concentrate their efforts on problematic areas. Providing not just dashboard reports but also detailed reports that can be exported into spreadsheets also proved very important for auditors, as they need to document and capture results in the final audit result documents, as well as in presentations.

For application owners the analysis based on the model gives continuous view of how controls are working in regards to the set security policies or SOX control requirements, allowing them to assess controls not just during audits. It also saves time required from application owners to prepare for audit since all data is already available in a suitable form for the auditors.

6. RELATED WORK

Our approach to assurance management fits within the overall risk lifecycle of identifying key business assets, vulnerabilities and then designing suitable control architecture. The approach is obviously complemented by tools that aid the risk analysis process and help model risks and vulnerabilities such as CORAS [11]. Tools such as Secure Tropos [12] provide a way of modelling the relationships and responsibilities between various actors within an environment and as such it could be applied to a service environment. However, it is important that there is ongoing assurance between parties and our assurance management approach provides a way of delving deeper into the operational aspects of operating a service.

The assurance management approach is highly related to the audit process where the model is used to automate much of the routine audit field work. A current tool popular with auditors is ACL (Audit control language) that provides a number of data analysis techniques and templates that auditors can apply. Our modeling approach can automate many of these tasks although there would be obvious advantages in integrating in audit tools to aid in the analysis and help present results in a familiar fashion. The approach of modeling a control set can be contrasted with light weight approach developed by HP Internal Audit [13] to find process based leading indicators and symptomatic lagging indicator. This approach captures auditors' knowledge and helps them focus their attention with a low cost. The modeling of the control architecture captures much more of the structure and can help communicate this structure and associated performance between service boundaries.

7. CONCLUSIONS

Compliance and heightened demands for improved corporate governance and fiscal transparency are not one-time events. Companies are increasingly calling on internal auditing to help improve performance by identifying areas of operational inefficiencies, risks in outsourcing environments, and fraud. The only way internal auditing can meet these demands — without growing its audit department significantly — is through the effective use of technology. In this paper we presented how model based assurance technology has been used within HP IT audits to model the internal controls and automate analysis on the effectiveness of the implemented controls in real IT environments.

The results of such analysis are useful not just to the auditors but also to the system/application owners. Continuous analysis of controls based on the created models provide for timely, sometimes immediate, identification of anomalies or control gaps, and, once these gaps are identified, actions can be taken by the system owners to identify and correct problems before they get out of control. Being used over longer periods continuous model-based control analysis can help validate the adequacy of the mitigating controls; help achieve an organizational culture where risks are taken seriously hence achieving greater effectiveness of the controls and ultimately better management of the risks.

8. REFERENCES

[1] Sarbanes Oxley Act, http://www.sarbanes-oxley.com

[2] Payment Card Industry Data Security Standard http://usa.visa.com/download/business/accepting_visa/ops_risk_management/cisp_PCI_Data_Security_Standard.pdf

[3] Health Insurance Portability and Accountability Act of 1996. http://www.legalarchiver.org/hipaa.htm

[4] Auditing Internal Control over Financial Reporting, Public Company Accounting Oversight Board – PCAOB.

[5] ITGI, Control Objectives for Information and Related Technologies (COBIT), 3rd edition, 1998.

[6] The HP IT Service Management (ITSM) Reference Model

[7] COSO: The Committee of Sponsoring Organisations of the Treadway Commission, http://www.coso.org

[8] *ACL Audit Analytics Technology*, ACL Services, http://www.acl.com/pdfs/ACL_Technology.pdf

[9] A Baldwin, Y Beres, and Simon Shiu. Trust Record: High Level Assurance and Compliance. In *Proceedings of the 3rd International Conference on Trust Management*, LNCS 3477, Paris, May 2005

[10] A Baldwin, Y Beres, and Simon Shiu. Using assurance models to aid the risk and governance lifecycle. In *BT Technology Journal*, Vol. 25 No. 1, January 2007.

[11] F Vraalsen, F den Braber, M Soldal Lund, K Stolen. The CORAS Tool for Security Risk Management. In *Proceedings 3rd International Conference on Trust Management*, IETF Vol 3477, 2005.

[12] P Giorgini, F Massacci, J Mylopoulos and N Zannone. Requirements Engineering Meets Trust Management: Model, Methodology and Reasoning. In *Proceedings 2nd International Conference on Trust Management*, 2004.

[13] B Ames *et al*. Continuous Control Monitoring: Enabling Rapid Response to Control Breakdowns. Audit Director Roundtable http://www.audit.executiveboard.com/, 2004.

[14] Symantec Enterprise Security Manager, Symantec Corp. http://www.symantec.com/enterprise/products/

[2] For the account termination control around 80% of time saved; for the approval process around 50% of time saved when the application matched the model requirements.