An Approach to Data-driven Detective Internal Controls for Process-aware Information Systems

Rafael Accorsi
University of Freiburg, Germany
accorsi@iig.uni-freiburg.de

ABSTRACT

This paper argues for an approach for the well-founded, scalable detective internal controls to assist controllers in swiftly and reliably identifying violations of control objectives in business process executions. Considering the usual internal control setting, in which controllers have a process and policy specification (target state) and the corresponding event log generated during the process execution (actual state), our approach automatically analyzes the entire set of process executions comprised in the event log. For this, novel, formal approaches to data-driven conformance checking need to be devised.

Categories and Subject Descriptors

K.6.5 [**Manage-ment of Computing and Information Systems**]: Security and Protection

Keywords

Detective internal control, Business Process Management, Usage control

1. INTRODUCTION

Detective controls are designed to identify, a posteriori, the violation of control objectives in enterprise information systems. Control objectives include, for instance, abuse of rights, conflict of interest and four-eye rule. Generally, such controls, as well as compliance rules, can be regarded as usage control requirements [3, 23, 26].

Despite the recent series of accounting failures and associated regulation efforts, detective internal control practices for business processes – and more generally process-aware information systems [19] – are still based upon the manual analysis of random sample logged process executions [36]. The resultant control risk is high, i.e. the probability and the associated costs of overlooking violations, thereby endorsing fraud or eventually failing an audit.

DUMW'11 Lyon, France.

The Société Générale incident is a particularly prominent, well-documented example to illustrate the impact of flawed internal controls. Unauthorized transitions by trader Jérôme Kerviel led to the loss of nearly 5 billion Euros. These transactions (e.g., directional bets concealed by fake portfolios) were only possible because internal controls, such as those for segregation of duties and abuse of rights, have been circumvented. These errors were not spotted despite the availability of complete logs, whereas the reason for this is faulty detective internal controls [20].

This paper argues for an approach for automated data-driven detective internal controls. We call it Adict. Adict builds upon conformance checking, i.e. analysis based upon comparisons between the target state (process specification) and the actual state (event logs). Conformance checking [34] is a technique within the field of process mining [33], which is employed to detect discrepancies between the target and the actual behavior. However, up to now only structural features of process runs, such as deviating executions and incidence of paths, could be detected. The consideration of more sophisticated control objectives or security policies (e.g., separation of duties and usage control requirements) are not possible.

Adict extends conformance checking to determine the compliance of event logs with various types of control objectives. Based upon colored Petri nets, which provide a suitable semantics for data-driven business process reasoning [8], the techniques to be developed in Adict address: (a) the declarative formalization of process-independent, semantically justified control objectives as Petri net anti-patterns, whereas specific places in these patterns denote control objective violations; (b) the analysis of process executions by replaying the event log traces in the process specification to detect violations of the patterns; and (c) the automated derivation and circumscription of need-to-know requirements based upon the target and the actual states.

The remainder of this paper is structured as follows. After a brief survey on related approaches, Section 2 gives an overview of Adict's approach and its main building blocks and Section 3 and indicates ongoing work.

Web as a more general application context.

While the techniques suggested in this paper are motivated by and shown in the context of internal control/auditing, we believe that they could be equally employed in other settings, in particular the web. In essence, Adict is an approach for detective usage control, i.e. a posteriori generation of compliance evidence with the designated policies.

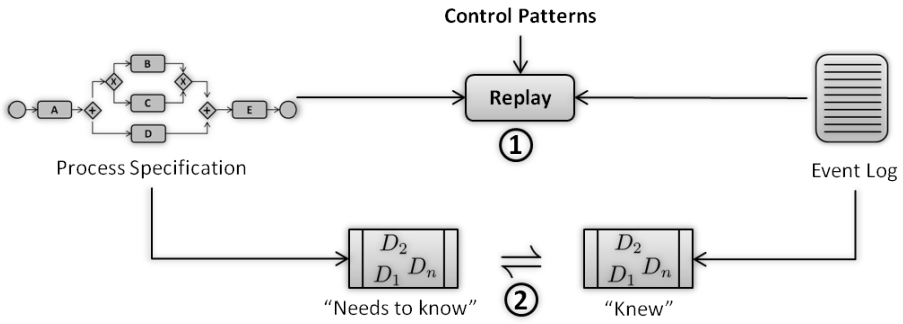

Figure 1: Overview of Adict.

One exemplary application could be a "Social network dashboard". Here, Adict could provide users with pre-defined patterns capturing usage control policies (e.g., on data retention and third-party notification), so that users could click their way through the policies. Upon request, a particular view of the log – corresponding to the user – could be generated and checked for compliance.

The realization of such a feature anticipates, one the one hand, the willingness of service providers to make their processes (at least in part) public. This could be achieved in situations where economical incentives to transparency were in place. On the other hand, it assumes a reliable infrastructure that includes secure logging and remote attestation technologies [2]. Otherwise the evidence generated during the check is void.

Related work.

We have amply investigated the use of conformance checking to conduct security and compliance audits in process-aware information systems [6]. This case study indicates that there are mechanisms to determine which traces in the event log fit into the model, as well as to perform straightforward compliance checks based solely upon the log files [1]. However, checking data-driven constraints, such as data propagation, is not possible. This appears to be a more general shortcoming, and in fact challenge, when it comes to analyzing log data "in the large" [24] and process mining.

Aalst et al. [35] present the Online Auditing Tool (OLAT) framework which in essence integrates the existing mechanisms to provide for continuous auditing and, in further stages, also preventive monitoring. However, in OLAT the actual state is not employed. In addition, only elementary control objectives can be considered. Accorsi et al. [4] develop a monitor architecture based upon conformance checking to identify and evaluate process deviations during its execution with regard to the compliance with security rules.

Some approaches to detective internal control rely solely upon event logs (e.g., [5, 13, 17]). They cannot be seen as conformance checking in the strict sense, as there is no consideration of the target state (in terms of process model) during the analysis.

2. APPROACH OVERVIEW

Figure 1 provides an overview of the Adict approach. It builds upon two types of conformance checking. Firstly, replays determine whether control objectives formalized as Petri net patterns, are violated. For this, a Petri net representation of the process is employed, on top of which traces are replayed. Similar to security automaton [29], whenever a replayed trace activates a pattern, the corresponding control is violated. (Note that patterns capturing the violation of a property are generally referred to as anti-patterns. For the sake of readability, below we simply refer to them as "patterns".) Secondly, Adict focuses on need to know requirements, which while relevant, are usually not considered or checked automatically. It employs abstractions to circumscribe, from the target state, the data set each subject needs to know (as for the process specification) and derives from the event log the data set knew by each subject; subsequent conformance checks with the data set "needed to know" and the data set "allowed to know" provide evidence on violations of abuse of rights (and hence "bad" information flows) and improper policy specification.

2.1 Business Process Specification

Business processes are traditionally specified using languages such as the Business Process Modeling and Notation (BPMN), Business Process Execution Language (BPEL) and Event-driven Process Chains (EPC). Generally, they have an execution semantics, but lack a formal semantics to allow the automated reasoning. The usual way to circumvent this problem is to map the specifications to Petri net models [32]. (Alternatively, process algebra can be employed. Still, the vast majority of approaches employ Petri nets.) Here, the process activities are considered transitions of the net and performing a transition consumes a token from one place, and produces the corresponding tokens on the other. The most notable dialect for this purpose is the Workflow net [32], which restricts the models regarding the form and transition semantics; for instance, that the net has unique and distinct start and end places, and that the black tokens are completely passed over during the execution.

However, Workflow nets do not allow for the representation of data items (and, more generally, resources). Correspondingly, approaches to mapping business process specifications to Petri nets consider only the structure and control flow of the process, not the exchanged data items. As a preparation for Adict, we devised a more expressive formalism called Information Flow Net (IFnet) [7]. IFnet combines colored Petri nets and Workflow Nets, and define mappings from BPEL and BPMN into IFnet models. (Definitions of soundness and the corresponding decidability results, adapted from Workflow nets, are available and hold.)

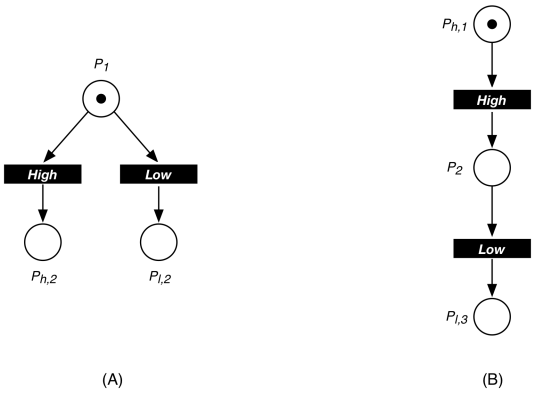

Figure 2: Patterns to capture interferences.

In doing so, Adict is able to reason about such data items, for instance, whether a data item moved down from a secret to a public domain.

2.2 Characterization of Control Objectives

Typical control objectives in process-aware information systems are [31]:

- *Four-eye principle*: business decisions and transactions need approval from two distinguished subjects prior to commitment.
- *Segregation of duties*: dissemination of activities and associated privileges for a specific business process among multiple subjects.
- *Binding of duties*: assignment of activities and associated privileges for a specific business process to one subject.
- *Conflict of interest*: subjects (and information) involved in the execution of one process should not be involved in the execution of another process.
- *Need-to-know*: subjects should only obtain the information necessary to run a specific process or carry out a particular task.

Generally, control objectives also comprise data usage requirements [12]; for example, that a data must be deleted after the execution of a process (data flow), or that an activity must be isolated from another set of activities (interference). Further, regulatory compliance requirements can be reduced to usage control requirements [26, 27] and, hence, be equally seen as control objectives.

Adict captures these control objectives as IFnet patterns in a way similar to [10]. In previous work, we employed such a patterns to provably capture particular information flow properties, in particular mandatory access control rules and different kinds of interferences, thereby extending "Place-based Non-Interference" [16]. To exemplify this specification style, the patterns in Fig. 2 capture a specific kind of non-interference, i.e. covert information flows. Specifically: assuming a multi-level security model [18], the patterns in Fig. 2 capture the Bisimulation-based Non-Deducibility on Composition (BNDC), which forbids a low subject from deriving information about high's behavior. At the conflict place P_1 in Fig. 2(a), high and low compete for the black token (control flow) and whenever high consumes the token, low can deduce high's action. At the causal place P_2 in Fig. 2(b) one action of high always follows one action of low. Overall, the control flow allows low to deduce information about high (interference), thereby violating, e.g., isolation.

For the moment, Adict has patterns to capture, for example, separation (and thus binding) of duties, conflict of interest and data flow requirements based upon mandatory access control.

2.3 Log Format

The log-based analysis of business process is generally referred to as process mining [33]. Process mining encompasses three types of approaches: *discovery* to reconstruct so-called de-facto process models from logs; *conformance* to check the extent to which the logs correspond to the original de-jure process; and based on such analysis, *enhancement* to improve the model in order to fulfill the expected properties. Adict focuses on conformance checking, as it builds upon comparing the actual and the target states.

The starting point for conformance checking is an event log. Each event in such a log refers to an activity (a well-defined step in some process) and is related to a particular case (a process instance). The events in a case are ordered and describe one "run" or "trace" of the process. Event logs also store supplementary information, such as the originator (person or device) triggering the activity, its role, the event's time stamp, required input and provided output. The following depicts the typical log format as input for process mining.

timestamp	activity	originator	input data	output data

The key assumption here is that the designated process-aware information systems provide for these fields, or that the corresponding log formats can at least be reduced to the format. There is enough evidence to substantiate this assumption [37]. Log formats, such as the eXtensible Event Stream (XES), allow for the realization of efficient mechanisms for log analysis, for instance process discovery and conformance checking.

2.4 Conformance Checking

In conformance checking, an existing process model is compared with an event log of the same process [34]. The comparison shows where the real (executed) process deviates from the modeled process. Moreover, it is possible to quantify the level of conformance and differences can be diagnosed. Conformance checking can be used to check if reality, as recorded in the log, conforms to the model and vice versa. There are various applications for this (compliance checking, auditing, Six Sigma, etc.). Adict exploits conformance checking for detective internal controls, which is in itself similar to an auditing setting (log analysis), even though under a different set of assumptions [31].

Conformance checking and performance analysis require an alignment of event log and process model, i.e., events in the event log need to be related to model elements and vice versa. Such an alignment shows how the event log corresponds to the process model. Assuming a business process specification and an event log as in Section 2.3, the central mechanism to check their conformance (or alignment)

consists of replaying the activities in the log into the corresponding activities of the process specification. Replay thus detects structural discrepancies between the target model and logs, in that traces, for example, execute activities in the wrong sequence or skip required transactions. Technically, considering a Petri net representation of the process, replay is realized as a Petri net "token game" [22] by forcing transitions to fire (if possible) in the order indicated by the trace.

Currently, conformance checks based upon replay address only structural aspects of the workflow. That is, data-driven checks (e.g., encompassing message passing or data exchange) is are possible; similarly, currently it is not possible to make the originator accountable for a certain task, as this information (log field) is not employed by the checks. Adict extends conformance checking with these dimensions. Furthermore, while triggering the activities, it also triggers the corresponding activities in the patterns that capture the control objectives. In doing so, the corresponding tokens are moved on in the control objectives patterns and, if applicable, indicate a violation whenever a "harmful" place is active.

2.5 Addressing Need to Know Requirements

The principle of "need to know" restricts the set of information that can be known by a subject to those data items strictly necessary to conduct the designated duties in a process. Need to know is closely related to the principle of "least privilege" [28] and often equated with it [15, 30]. This principle suggests that each subject in a system should be granted the most restrictive set of privileges (or the lowest "clearance") needed for the performance of authorized tasks.

Although related, least privilege and need to know focus on different aspects and, hence, require different mechanisms. While the former focuses on the rights, the latter focuses on the (maximal) set of data which can be accessed by a subject. Because in enterprises and corresponding process-aware information systems roles (and thereby subjects) usually possess more rights than those needed for the execution of a particular process [14], cascading accesses may take place [11], vulnerabilities exist [21] and break-glass policies allow for the temporary elevation of rights [25], detective internal controls must check whether these situations led to an abuse of rights in which a subject obtains more information than possible. Further, they should indicate whether covert access may have led to information gain. Put another way: rather than focusing solely on the rights, need to know must focus on the information that potentially flows to the subject.

Adict employs abstraction techniques to characterize, based upon the process specification, the set of data a subject "needs" to know in order to conduct a process. Similarly, abstraction, will be employed to obtain the set of information that such a subject "knew", together with possible interferences (Section 2.2). This allows Adict to detect discrepancies between these two "epistemic" states and, consequently, identify violations of control objectives, as well as abuse of rights.

3. SUMMARY

This paper argues for several types of conformance checking for the automatic detective internal controls. The goal is to improve the quality of process-aware information systems by reliably and timely detecting violations and, thereby, allow the enhancement of process design (or execution engine). The Adict approach provides for declarative, process-independent characterizations of control objectives that can be straightforwardly mapped to process-specific patterns and, subsequently, serve as a basis for conformance checking. Given that, replays attempt to reproduce the traces into the model, simultaneously triggering the corresponding patterns. Further, Adict addresses the characterization and detection of need to know requirements in the context of business processes. To the best of our knowledge, this is a novel, promising of reasoning about automated detective controls. We have carried out experiments with a prototypical implementation focusing solely on properties encoded on solely structural patterns. (To this end, we synthesized log files with SWAT, the Security Workflow Analysis Toolkit [9].) While the Python implementation detect all the violations in the log, it took around a minute to traverse an event log with 500K cases. (We employed in the test a virtual machine with Ubuntu 10.10 64-Bit, 4GB RAM and one core with 2,67 GHz). Further optimizations are possible.

A particular attractive feature on Adict is that it allows the quantification of incidents. The fact that the log "chunk" designates the particular view of the reality that held for a time period makes it possible to determine, for the designated time period, for instance the amount of information that flows over a covert-channel. We will exploit this dimension in a future time point.

4. REFERENCES

[1] R. Accorsi. Automated privacy audits to complement the notion of control for identity management. In E. de Leeuw, S. Fischer-Hübner, J. Tseng, and J. Borking, editors, *Policies and Research in Identity Management*, volume 261 of *IFIP Conference Proceedings*, pages 39–48. Springer, 2008.

[2] R. Accorsi. Log data as digital evidence: What secure logging protocols have to offer? In *Proceedings of the 1st IEEE Workshop on Computer Forensics in Software Engineering*, pages 398–403. IEEE Computer Society, 2009.

[3] R. Accorsi, L. Lowis, and Y. Sato. Automatisierte compliance-zertifizierung cloud-basierter geschäftsprozesse. *Wirtschaftsinformatik*, 53(3):139–149, 2011.

[4] R. Accorsi, Y. Sato, and S. Kai. Compliance monitor for early warning risk determination. *Wirtschaftsinformatik*, 50(5):375–382, October 2008.

[5] R. Accorsi and T. Stocker. Automated privacy audits based on pruning of log data. In *Proceedings of the EDOC International Workshop on Security and Privacy in Enterprise Computing*. IEEE, 2008.

[6] R. Accorsi and T. Stocker. On the exploitation of process mining for security audits: The conformance checking case. In *ACM Symposium on Applied Computing*, 2012.

[7] R. Accorsi and C. Wonnemann. Strong non-leak guarantees for workflow models. In *ACM Symposium on Applied Computing*, pages 308–314. ACM, 2011.

[8] R. Accorsi and C. Wonnemann. InDico: Information flow analysis of business processes for confidentiality requirements. In J. C. et al., editor, *ERCIM Workshop on Security and Trust Management*, volume 6710 of

Lecture Notes in Computer Science, pages 194–209. Springer, 2011.

[9] R. Accorsi, C. Wonnemann, and S. Dochow. SWAT: A security workflow toolkit for reliably secure process-aware information systems. In *Conference on Availability, Reliability and Security*, pages 692–697. IEEE, 2011.

[10] N. Adam, V. Atluri, and W.-K. Huang. Modeling and analysis of workflows using petri nets. *Journal of Intelligent Information Systems*, 10(2):131–158, 1998.

[11] R. Anderson. *Security Engineering*. Wiley, 2nd edition, 2008.

[12] V. Atluri and J. Warner. Security for workflow systems. In M. Gertz and S. Jajodia, editors, *Handbook of Database Security*, pages 213–230. Springer, 2008.

[13] D. A. Basin, M. Harvan, F. Klaedtke, and E. Zalinescu. Monitoring usage-control policies in distributed systems. In C. Combi, M. Leucker, and F. Wolter, editors, *Symposium on Temporal Representation and Reasoning*, pages 88–95. IEEE, 2011.

[14] M. Benantar. *Access Control Systems*. Springer, 2006.

[15] J. Biskup. *Security in Computing Systems - Challenges, Approaches and Solutions*. Springer, 2009.

[16] N. Busi and R. Gorrieri. Structural non-interference in elementary and trace nets. *Mathematical Structures in Computer Science*, 19(6):1065–1090, 2009.

[17] J. Cederquist, R. Corin, M. Dekker, S. Etalle, J. den Hartog, and G. Lenzini. Audit-based compliance control. *International Journal of Information Security*, 6(2-3):133–151, 2007.

[18] D. Denning. A lattice model of secure information flow. *Communications of the ACM*, 19(5):236–243, 1976.

[19] M. Dumas, W. van der Aalst, and A. ter Hofstede, editors. *Process-aware Information Systems: Bridging Poeple and Software through Process Technology*. Wiley, 2005.

[20] J. Epstein. Security lessons learned from Société Générale. *IEEE Security & Privacy*, 6(3):80–82, 2008.

[21] L. Lowis and R. Accorsi. Finding vulnerabilities in SOA-based business processes. *IEEE Transactions on Service Computing*, 4(3):230–242, 2011.

[22] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[23] K. Namiri and N. Stojanovic. Using control patterns in business processes compliance. In M. Weske, M.-S. Hacid, and C. Godart, editors, *Proceedings of the International Workshop on Web Information Systems Engineering*, volume 4832 of *Lecture Notes in Computer Science*, pages 178–190. Springer, 2007.

[24] A. J. Oliner, A. Ganapathi, and W. Xu. Advances and challenges in log analysis. *Communications of the ACM*, 55(2):55–61, 2012.

[25] D. Povey. Optimistic security: A new access control paradigm. In *Proceedings of the New Security Paradigm Workshop*, pages 40–45. ACM Press, 1999.

[26] A. Pretschner, F. Massacci, and M. Hilty. Usage control in service-oriented architectures. In C. Lambrinoudakis, G. Pernul, and A. M. Tjoa, editors, *Proceedings of the 4th International Conference on Trust, Privacy and Security in Digital Business*, volume 4657 of *Lecture Notes in Computer Science*, pages 83–93. Springer, 2007.

[27] S. Sackmann and M. Kähmer. ExPDT: A policy-based approach for automating compliance. *Wirtschaftsinformatik*, 50(5):366–374, October 2008.

[28] J. Saltzer and M. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, September 1975.

[29] F. Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1):30–50, February 2000.

[30] F. B. Schneider. Least privilege and more. *IEEE Security & Privacy*, 1(5):55–59, 2003.

[31] E. Vaassen, R. Meuwissen, and C. Schelleman. *Accounting Information Systems and Internal Control*. Wiley, 2010.

[32] W. van der Aalst. The application of petri nets to workflow management. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1998.

[33] W. van der Aalst. *Process Mining – Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.

[34] W. van der Aalst, A. Adriansyah, and B. van Dongen. Replaying history on process models for conformance checking and performance analysis. *Data Mining and Knowledge Discovery*, 2012.

[35] W. van der Aalst, K. van Hee, J. M. van der Werf, A. Kumar, and M. Verdonk. Conceptual model for online auditing. *Decision Support Systems*, 50(3):636–647, 2011.

[36] W. van der Aalst, K. van Hee, J. M. van der Werf, and M. Verdonk. Auditing 2.0: Using process mining to support tomorrow's auditor. *IEEE Computer*, 43(3):90–93, 2010.

[37] H. M. Verbeek, C. A. M. Joos, B. Dongen, and W. van der Aalst. XES, XESame, and ProM 6. In W. Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, P. Soffer, and E. Proper, editors, *Information Systems Evolution*, volume 72 of *Lecture Notes in Business Information Processing*, pages 60–75. Springer, 2011.