KU LEUVEN

FACULTY OF ECONOMICS AND BUSINESS

Advances in rule-based process mining: applications for enterprise risk management and auditing

Filip Caron, Jan Vanthienen, Bart Baesens

KBI_1305

Advances in Rule-Based Process Mining: Applications for Enterprise Risk Management and Auditing

Filip Caron[a,*], Jan Vanthienen[a], Bart Baesens[a,b,c]

[a] *Department of Decision Sciences and Information Management, Faculty of Business and Economics, KU Leuven, Naamsestraat 69, B-3000 Leuven, Belgium*
[b] *Vlerick Leuven Gent Management School, Vlamingenstraat 38, B-3000 Leuven, Belgium*
[c] *School of Management, University of Southampton, Highfield Southampton, SO17 1BJ, United Kingdom*

Abstract

Process mining research has mainly focused on the development of process mining techniques, with process discovery algorithms in the center of attention. However, far less research attention has been paid to the actual applicability of these process mining techniques in common business settings. Consequently, there only exists a partial fit between the existing process mining techniques and the compliance checking & risk management applications.

This research report contributes to the process mining and compliance checking research by proposing an effective and efficient rule-based approach for analyzing organizational information and processes. Additionally, a general content-based business rule taxonomy has been developed as a source of business rules for the compliance checking approach. Furthermore, we also provide formal grounding for and an evaluation of the rule-based approach.

Keywords: Business Rules, Compliance Checking, Risk Management, Process Mining, Process-Aware Information Systems

1. Introduction

While the value creation abilities of an organization are increasingly determined by the flexibility of their information systems and business processes, this flexibility may also pose significant risks that could have an enormous impact on achieving the corporate objectives [56]. Additionally, contemporary organizations are faced with evermore restricting directives, both external (e.g. government and trade associations) and internal (e.g. business policies and corporate social responsibility programs). Consequently, the organization's management performs a risk assessment and implements appropriate risk responses, e.g. control procedures. Well-known examples of control procedures include the segregation of duties, the authorization rules and the inclusion of approval activities.

*Corresponding author, Telephone: +32 16 32 65 58, Fax: +32 16 32 66 24
Email address: filip.caron@econ.kuleuven.be (Filip Caron)

Shareholders and other stakeholders of the organizations will demand an independent assessment of the effectiveness of these risk responses, which is typically performed by auditors. Both the risk and control effectiveness assessments make use of (similar) compliance checking techniques.

Several research contributions have suggested the use of process mining techniques for such an effectiveness assessment, i.e. to uncover potential compliance failures (e.g. [62, 38, 53]). Process mining refers to the set of techniques that analyzes event logs of process-aware information systems to acquire unique insights into the real processes [73, 69]. These event logs contain an untapped reservoir of knowledge about the business operations, detailed information on the business events (e.g. timestamps, case identifiers, originator identifiers, etc.) is recorded in a structured way.

While compliance checking in the audit department has been suggested as a potential application for several process mining techniques, much less attention has been paid to fully adapting the techniques to the audit's specific needs. Furthermore, other governance, risk and compliance activities are not taken into account. Moreover, the description of specific applications of process mining in a control (assessment) setting are rare and potential generalizations are not provided. This research report contributes to the applied process mining research by:

- Proposing an effective and efficient rule-based compliance checking approach with process mining for analyzing organizational information and processes, with concrete applications in both (audit) compliance checking and risk management.
- Presenting a general content-based business rule taxonomy, resulting in a set of frequently used rule patterns for control (assessment) and risk identification & assessment.
- Providing both a formal grounding for and an evaluation of the rule-based compliance checking with process mining.

The outline of the research report is as follows: section 2 provides an overview of compliance checking with process mining, describes the partial/limited fit and introduces the rule-based compliance approach. Section 3 presents a general content-based rule taxonomy. In the body of the research report we discuss the rule-based compliance checking approach with process mining: the details and formal grounding in section 4, the identified rule patterns in section 5 and an evaluation of the opportunities and challenges in section 6. The final section concludes the research report and presents an outlook for future research in this area.

2. Compliance Checking with Process Mining

The study of compliance checking is an emerging research field which also holds important challenges for the process-aware information systems. The major contributions of compliance research in process-aware information systems can be grouped along the different phases of the business process management lifecycle, e.g. [43, 60, 42, 81]. However, this contribution will focus on the monitoring and analysis phase and more specifically on process mining as a set of techniques for compliance checking.

2.1. Process Mining

Process mining addresses the problem that most organizations, and consequently also the persons with compliance checking and risk management activities, have very limited information about what is actually happening in the business processes [73, 78, 25]. Insights into the real behavior are acquired through analysis of the structured information contained in information systems' event logs. Process mining has received a vast amount of research attention resulting in a plethora of techniques, e.g. process discovery techniques [9, 16, 24, 63, 79, 33], techniques for the analysis of event log data [6, 64, 65, 68], techniques for trace classifications [58, 20], process metrics [53, 18] and applied research [67, 44]. Figure 1 schematically represents the process mining architecture.

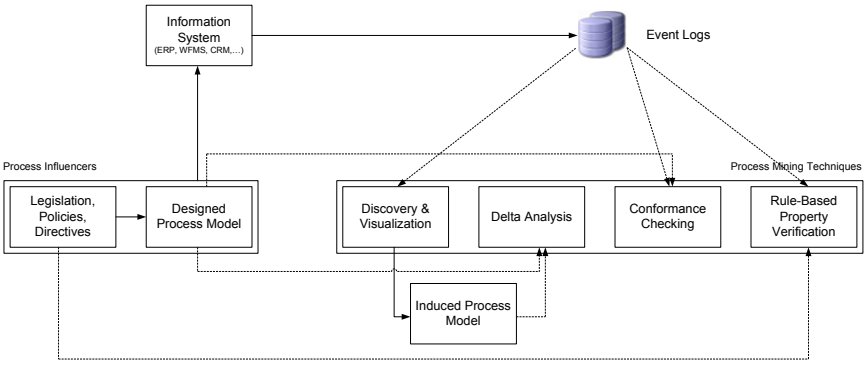

Figure 1: Process Mining Architecture

2.2. Traditional Compliance Checking with Process Mining

Three broad sets of techniques are of main interest for compliance checking: process discovery & visualization, conformance checking & delta analysis and rule based process mining.

The set of ***process discovery and visualization techniques*** enables the acquisition of a *full and precise insight in the real business process, summarized in one visual.* Therefore process visualization can be considered as the ideal technique for process exploration and thus a first step in a compliance analysis [62]. In a compliance checking setting the α [63], the $\alpha++$ [16] and the heuristics miner [79] (only in combination with low thresholds) are suitable to *obtain a process model* from the event log of an information system. Whereas process visualization traditionally focused on the overall control-flow, the *visualization of other business process aspects* can be of benefit for a compliance analyst, e.g. the performance sequence diagram analysis plug-in [70], the originator-by-task matrix [71], bottleneck analysis [54]. Process visualization as an auditing tool is advocated, for example, in [37, 62, 1].

Conformance checking and delta analysis both aim at detecting *inconsistencies between a prescriptive process model and its corresponding real-life process.* The major difference between both lies in the comparison base for the real-life process, conformance checking uses the event log while delta analysis uses a derived process model [52, 53,

3, 10, 32]. The execution paths specified in the process model often contain multiple *implicit internal controls* (e.g. approval is needed before paying a bill). Both the risk assessment and the compliance analysis solely rely on the *recall/fitness* dimension, which describes the extent to which the behavior in the event log can be associated with valid execution paths in the prescriptive process model.

Rule-based property verification approaches allow the analyst to verify *specific questions* [65]. Examples of properties an analyst might want to check include the four-eyes principle and a set of ordering constraints. LTL-Checker plug-in [14] for ProM provides the analyst with sixty *application-independent and configurable templates* of common process properties (e.g. the existence of certain activities). Semantic LTL-Checker adds a functionality that allows the user to provide concepts as input to the parameters of LTL formulae [15]. SCIFF Checker uses configurable CLIMB rule templates to verify process properties [45]. Several research contribution successfully applied property verification based on the standard templates stored in the LTL-Checker plug-in, e.g. [37, 38, 74, 6].

2.3. Partial fit between Compliance Checking and Existing Process Mining Techniques

Existing process mining techniques are often ***not fully adapted to the risk management and audit compliance checking*** setting. Process mining researchers have primarily focused on improving techniques for *control-flow* and to a far lesser extent for social/organizational analysis. Besides the obvious requirement to be able to analyze and perform controls on the additional data in the event logs (e.g. amounts), this research focus results in other important restrictions. Firstly, conformance metrics might provide *only a first impression* of the overall conformance with the designed model. An in-depth evaluation of the problematic process parts, where the deviation from the designed model is significant, will be required. Additionally, we can *question the correctness of the designed business processes.* Procedural business process models often contain control-flow dependencies that are not dictated by internal or external directives, i.e. overspecification of the process model [47]. Consequently, process deviations affecting the conformance measures, do not necessarily violate any internal or external directive. Moreover, defining all possible execution paths to deal with natural variations in a business environment can be challenging.

Due to the assumption that an event log will not contain all possible behavior, process discovery and visualization techniques have to *balance model precision and generality.* However, in a risk management and compliance checking setting generality is not important, it might might actually result in a cover up of (infrequent) harmful process deviations.

2.4. Comprehensive Rule-Based Compliance Checking with Process Mining

To deal with the partial/limited fit between compliance checking and traditional process mining techniques, this research report will propose a comprehensive rule-based approach. The compliance analyst will be presented with an extensive set of configurable business rule patterns that can be used to describe the most common controls. These patterns can be tested against a broad set of process information, including the event log of at least one business process. Other data sources may include the human resource data bases, the client data bases, etc. By taking into account multiple business process event logs, the compliance analyst will be able to make reconciliations.

In addition to the ability to take into account specific case data of a variety of data sources, this rule-based approach to compliance checking with process mining will be robust to noise and consequently a trade-off between generality and specificity will not be made. As a result there will be no cover up of harmful low frequency process deviations. Furthermore, the approach enables the analyst to obtain a precise indicator of the effectiveness of the control or potential risk event. Finally, this compliance checking approach will not be confronted with the disadvantages of process overspecification.

A more elaborate discussion of the approach can be found in section 4. The next section will focus on identifying the business rule categories that can be used for the rule-based compliance checking approach.

3. Business Rules as a Resource for Compliance Checking with Process Mining

The Business rules research has mostly focused on improving the business rule languages in terms of expressibility [31, 30], comprehensibility [77, 5] and formalisms [26, 65]. Moreover, several contributions have indicated the relevance of business rules for modeling directives and control objectives, e.g. [55]. This section discusses the existing business rule classifications and proposes a new content-based taxonomy to fit a rule-based compliance checking approach used by business users.

3.1. Existing Business Rule Classifications

As a consequence of the increasing research attention for business rules, several business rule classifications were presented. Due to the fact that most of these classifications target a different audience (e.g. data base administrators, application developers, process professionals), they tend to use different bases of comparison.

The *BRS Rule Classification Scheme* proposes a categorization based on the three ***possible ways to react to an event***, i.e. reject, produce and project [51]. The rejectors, typically called constraints, are business rules that tend to disallow an event if this would create a violation of the rule. Producer business rules automatically measure something, compute or derive a specific value or infer a term or fact. Projector business rules automatically invoke specific actions when relevant events occur.

In its GUIDE report the *Business Rules Group* distinguishes three main business rule categories based on ***structural aspects***, i.e. derivation, structural assertion and action assertion [28]. Derivation business rules define how additional knowledge can be obtained from existing knowledge. Structural assertions can be either the definition of a business term or a fact relating different terms to each other. The third category, the action assertions, deals with constraining enterprise behavior in some way, examples include authorizations and integrity constraints. Similarly, *Ross* distinguishes the definitional rules from the behavioral rules, which are business rules that can be violated directly (comparable to action assertions) [50].

According to *Wagner* rules can be classified by their ***format*** and the elements from which they are composed [77]. Derivation rules consist of one or more conditions and one or more conclusions, whereas production rules combine one or more conditions with one or multiple actions. Integrity rules or constraints are composed of a constraint modality and a constraint assertion. Every reaction rules contains a triggering event, an optional

condition and a triggered action and/or post-condition. Finally, transformation rules focus on controlling the change of a system's state.

Declarative process modeling contributions (e.g. ConDec [47]) implicitly provide a classification structure for ***process or control-flow oriented*** business rules. Among these business rules we distinguish types such as activity existence rules, relationship rules and choice rules. The existence rules deal with the cardinality of one activity type within a process. While required order between activities can be defined with relationship rules, the choice rules specify the necessity to choose between multiple activities.

Comparably, *data modeling and data base* contributions have been structuring the ***data oriented business rules***. The Object-Role-Modeling methodology [31], for example, specifies different role-based fact types for typical data rules such as existential facts, elementary facts, uniqueness constraints, mandatory roles, value constraints, derivations and ring constraints. In [13] a distinction is made between constraints and derivations. The constraints are further divided in state, transition and stimulus/response constraints, specifying respectively the legal values, the allowed value changes and a combination of triggering event and action. The derivations on the other hand can be subdivided into inferences and computations.

Finally, multiple contributions propose the ***modality or level of enforcement*** used for a business rule as a potential classification basis, e.g. [29, 31]. While alethic business rules must be satisfied, deontic rules are obligatory but can be violated (in extreme cases they can be considered as guidelines).

3.2. Limited Business Audience for Existing Classifications

The useability of a specific business rule classification scheme will be primarily determined by its intended audience. While most of the existing business rule classifications will focus on technical aspects, this research report aims at providing adequate guidance to the business users that need to translate regulations, directives and other elements of the business environment into business rules. Therefore the business rule taxonomy presented in section 3.3 and further elaborated for a process mining setting in section 4 **classifies business rules based on their content type**. Classifying business rules based on content and adding a template to each individual type, which is both understandable and grounded in a formal definition, enables business users to perform controls from a broad spectrum of common types.

3.3. Business Rule Content Taxonomy

Every business rule type deals with an abstract business matter (e.g. an activity authorization), it has a **content** type. For use within a specific corporate setting, a specific business **context** should be augmented (e.g. a manager is authorized to perform a sign/approve activity). The business rule taxonomy presented in figure 2, groups the most common business rule types in contemporary organizations by their content type. Four main categories have been distinguished: *concept-oriented business rules, fact-oriented business rules, activity-oriented business rules* and *organizational-oriented business rules.*

Concept-oriented rules are business rules that specify something of importance to the business. While common concepts (e.g. customer) are generally understood, other more specific business concepts require a *concept definition* (e.g. gold customer) [28].

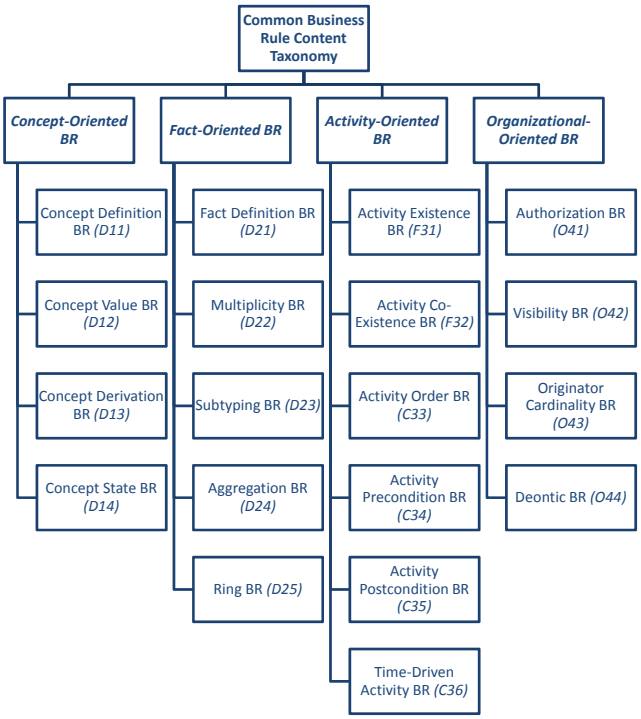

Figure 2: Common Business Rule Content Taxonomy

The *concept value* and *concept derivation* business rule types group all the business rules that define restrictions to the acceptable value. Concept derivations specify allowable value sets, intervals or uniqueness constraints, whereas concept derivations specify how the value of a concept can be inferred/computed from other concepts [13]. *Concept state* business rules on the other hand can specify both the initial concept state and the possible state transitions.

Secondly, relationships between concepts are specified and constrained by **fact oriented business rules**. Comparable to concepts, context specific facts require exact *fact definitions*. *Multiplicity business rules* restricts the number of instances for a concept type in a specific fact type, e.g. each American has exactly one social security number. The *subtyping business rules* group all business rules that specify the aspects of working with different subclasses, i.e. subtyping definitions, exclusiveness constraints and exhaustiveness constraints. *Aggregation business rules* group the business rules that specify different types of 'is part of' or 'is composed of' relationships between different concepts. *Ring business rules* are rules that specifies restrictions on a pair of compatible concepts (i.e. same concepts or concepts with one or more subtyping business rules) in a fact type. A typical example of a ring business rule is 'if employee$_1$ supervises employee$_2$ than it is impossible that employee$_2$ supervises employee$_1$'. The fact-oriented business rules are related to the ones typically discussed in data modeling, e.g. [31].

The third set of business rules is composed of **activity-oriented rules**, all focusing

on different aspects of the activities which are part of the business operations. *Activity existence business rules* define restrictions on the (number of) occurrence(s) of a specific activity in the context of one business case. Comparably, *activity co-existence business rules* restrict the co-occurrence of multiple activities within the context of a specific business case. *Activity order business rules* define sequence restrictions on multiple activities. The previous activity-oriented business rules are heavily related to the set presented in [48]. *Activity preconditions* and *activity postconditions* are business rules that specify a condition that needs to be fulfilled respectively before and after the execution of a specific activity. Time-driven activity rules impose a time restriction on the business operations, on the activity duration or on the time interval between particular activities.

Finally, the **organizational-oriented business rules** are business rules that model the obligations, restrictions, permissions etc. of specific agents within the organization. Different types of agents can be defined ranging from a specific employee or role to entire divisions. *Authorization rules* specify whether a particular type of agent is allowed to perform a particular type of activity [61]. Comparably, *visibility rules* define if is an agent of a particular agent type may access particular information. How many times a (particular) agent of an agent type may execute a specific activity will be recorded with an *originator cardinality business rule. Deontic business rules* specify when deontic assignments come into existence or cease to exist based on the occurrence/absence of certain business events [21]. Often deontic business rules are combined with a time-restriction.

4. Comprehensive Rule-Based Compliance Checking with Process Mining

The previous sections provide both the legitimation for a rule-based compliance checking approach with process mining and an overview of potentially interesting business rule types. This section will start with an elaborate discussion of the approach. Followed by an adaption of the general content-based rule taxonomy and a formal foundation for the approach. Finally, some possible extensions including are discussed (i.e. scope, duality of activities and artifacts and interprocess reconciliations).

4.1. Architecture behind Comprehensive Rule-Based Compliance Checking with Process Mining

The comprehensive rule-based compliance checking approach with process mining is based on an architecture consisting of three main building blocks: the 'business provenance' (or process information) block, the 'regulation, policies & directives' block and the 'techniques' block.

Business provenance deals with the systematic and reliant recording of business events and (evolutions of) other business artifacts, it keeps track of both the current status and the history. As business processes are becoming heavily supported by process-aware information systems, the event logs of these information systems become a valuable source of information on the business operations. This process centric information can be further enriched with information from other data sources, such as the customer management systems with the client base, the human resource data, etc. The accurate tracking as proposed by business provenance techniques is essential for compliance checking and will enable root cause analyses for compliance failures.

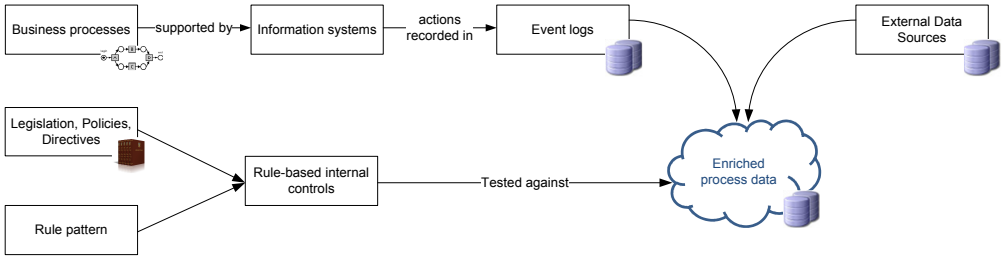

Figure 3: Advanced rule-based process mining architecture

Secondly, **legislation, policies and other directives** are the major purposes that justify the existence compliance checking. These directives constrain the business operations of an organization. Their origins can be structurally different, varying from external rules imposed by the government (e.g. Sarbanes-Oxley act), over standards set by trade associations, to internal policies specified in support of the organizations corporate social responsibility programs.

The third architectural building block contains the **techniques** that are used to perform compliance checking: configurable business rule patterns and process mining. The directives, presented in the previous building block, form a control objective that is translated into one or more specific controls. These controls can often be easily mapped on a set of business rules. Moreover, while the controls are devised for a specific organizational setting, it can be concluded that most of them follow one specific rule structure (e.g. segregation of duties, activity existence patterns, arithmetic derivation patterns, etc.) from a limited rule structure set. For these frequently reoccurring rule structures we will define a rule pattern which is easily understandable by business users, and can be easily configured to the specific setting. Additionally, each rule pattern is mapped on a formal specification which allows an automated rule testing against the enriched process data through process mining based compliance checkers, i.e. (process) event-oriented compliance checkers such as the LTL-checker or any other query environment.

4.2. Adapting Business Rule Content Taxonomy for Process Mining

The advanced rule-based process mining architecture indicates the necessity of a comprehensive set of configurable business rule patterns. While the common business rule content taxonomy presented in section 3.3 forms a strong base for this pattern set, we have to argue that the taxonomy is not adapted to the typical event-oriented structure of an event log. Therefore this section adapts the classification structure to perfectly fit the proposed rule-based compliance checking approach with process mining. The classification structure is based on **two dimensions**: the process mining perspective and the rule restriction focus.

4.2.1. Process Mining Perspective Dimension

The first dimension of the classification refers to the **process mining perspective (PMP)** that is used in the business rule. Four different perspectives on business process modeling were introduced in [12] and can also be used to classify the business rule types in this context.

- *Functional process perspective (PMP1)* that deals with the process elements (e.g. activities) that are being performed/occur in a process instance, as well as the relevant process artifacts linked to these process elements (e.g. an invoice artifact for a pay activity). The business rules in this class form a subset of the business rules in the F31 and F32 classes in the general taxonomy.
- *Control-flow process perspective (PMP2)* that covers the process behavior in terms of when process elements can be performed/occur in a process instance. This includes a wide variety of ordering relations between processes elements, as well as complex decision making conditions, entry and exit criteria, etc. Control-flow business rules could be classified in either class C33, C34, C35 or C36.
- *Organizational process perspective (PMP3)* that focuses on the organization behind the business process, which agent performs the different process elements in a process instance taking into account factors such as timing, environmental conditions, etc. Note that agent can have a broad interpretation, varying from a single person, over a department to a whole organization. This perspective is represented by the organizational-oriented business rule classes in the general taxonomy.
- *Data process perspective (also known as informational perspective) (PMP4)* that represents the informational elements (e.g. event data, case date, etc.) that are used, produced or manipulated during the process, as well as relationships among them. Business rules from this perspective belong to either the concept-oriented or the fact-oriented business rules.

These process mining perspectives can be grouped by their ***data requirements*** for event logs. For business rules from the functional and control-flow it suffices that the event log contains timestamps, event and activity identifiers. The organizational and data perspective based business rules require additional event data, such as an originator identifier, an amount, etc.

4.2.2. Rule Restriction Focus Dimension

Secondly business rules can be classified along their main **rule restriction focus (RRF)**. Five new and distinctive business rule restriction focuses are identified:

- *Cardinality-based rules (RRF1)* are business rules that restrict the number of allowed instances of a specific process element type (i.e. activities, process events, originators or other relevant event data) in a specific process instance.
- *Coexistence rules (RRF2)* can be defined as business rules that restrict the coexistence of process elements of different types over the execution of a specific process instance.
- *Dynamic data-driven rules (RRF3)* specify the influence of certain data elements (i.e. case or event data) and their value on the occurrence of process elements in a specific process instance.
- *Relative time rules (RRF4)* focus on specifying a time restriction on process elements relative to certain points in a process execution (e.g. start of a process, completion of a specific activity, etc.).
- *Static property rules (RRF5)* deal with specifying a specific property for a particular type of process element at a predefined process state.

While the first four rule restriction focuses deal with ***dynamic*** properties (i.e. history-based or future constraining), the last focus deals with ***static*** properties (i.e. properties in one specific process state). Both dimensions cover the wide spectrum of controls that can be implemented and evaluated with process mining techniques.

4.3. Formal Specification of Rule Patterns

While the business rule patterns are proposed with a need for comprehensibility by the business actors in mind, specifying them unambiguously is crucial in a risk management and compliance checking setting. An **unambiguous interpretation** of the business rule patterns and consequently the specific configurations, can be obtained by formally specifying them. Moreover, the existing model checking approaches enable an automated identification of conflicting business rules and consequently conflicting controls. Since each formal expression language has different semantics and a different expressive power, the definition of certain rule patterns may be unnaturally or even impossible in a specific formal specification language. Therefore the use of LTL is recommended for dynamic properties, whereas first order logic is more suited for static properties. This section starts with some preliminaries, followed by a discussion of the different uses of formal languages and ends with some concluding remarks.

4.3.1. Preliminaries: Specifying Business Processes, Business Events and Audit Trails

In this subsection we provide a formal definition for some basic concepts needed to perform advanced rule-based process mining, i.e. business process, business event and audit trail.

Definition 1: Business processes. A business process can be formally represented by a *process schema S*, which is defined by the tuple $(\mathcal{A}, \mathcal{R}, \mathcal{O}, \mathcal{P}, \delta, \pi, \mathcal{BR})$ where

- The basic constructs include: $\mathcal{A} = \{a_1, a_2, a_3, ..., a_n\}$ that denotes the finite set of all activities, $\mathcal{R} = \{r_1, r_2, r_3, ..., r_n\}$ that is used to refer the finite set of roles, $\mathcal{O} = \{o_1, o_2, o_3, ..., o_n\}$ that represents the finite set of originators and $\mathcal{P} = \{p_1, p_2, p_3, ..., p_n\}$ that stands for the finite set of (all other) properties.
- δ represents the property-type assignment function, $\delta : \mathcal{P} \rightarrow \Delta$. A possible set of generally relevant property types is defined as $\Delta = \{$*Time, InstanceIdentifier, Activity, EventType, Originator, Role, String, Rational, Boolean*$\}$.
- π is the property-set function that specifies the set of properties that is applicable for each activity, $\pi : A \rightarrow 2^{\mathcal{P}}$.
- $\mathcal{BR}$ the set of business rules specifying the relevant relations and constraints for a business process (e.g. precedence relations, required roles, etc.). These business rules may be either implicit (e.g. implicit preconditions in petri nets) or explicit (e.g. precondition in a declartive ConDec model).

During the execution of a business process a multitude of business events can be observed. A business event is a relevant occurrence of something (e.g. start of a specific activity) that happens at a specific time and is of special interest to the business.

Definition 2: Business event. The events related to activities, which are considered as the states in a process instance, are described as follows:

- An event is specified by the values of the related relevant data properties, as such an event can be denoted as $e : \mathcal{P} \rightharpoonup \mathcal{V}$ with $e \in \mathcal{E} = \{e_1, e_2, e_3, ..., e_n\}$ the set of all events and $\mathcal{V}$ the set of all possible values for the properties.
- The partial function at least assigns a value to the property activity, which indicates the activity related to the event, and all the related properties identified by the $\pi(a_i)$, in accordance with the property-type assignment.

Contemporary information systems store a multitude of information about these events in a structured way. Therefore, the resulting event logs (denoted by α, β, etc.) precisely describes the execution of all process instances, within a certain timeframe, i.e. the audit trail.

Definition 3: Audit trail. An audit trail $\sigma \in \mathcal{E}^*$ is an event sequence, where $\mathcal{E}^*$ represents all traces composed of zero or more events of $\mathcal{E}$. Additional properties:

- $\sigma = \langle e_1, e_2, ..., e_n \rangle$ denotes an audit trail with $|\sigma| = n$ the length of the trail and $e_i = \sigma[i]$ (with $1 \leq i \leq n$) the i-th event in the audit trail.
- $\sigma^{i\rightarrowtail}$ represents the suffix of σ starting at $\sigma[i]$, consequently $\sigma^{i\rightarrowtail} = \langle e_i, e_{i+1}, ..., e_n \rangle$

Each event log record contains the information of a specific event, including a specific event identifier $\iota = \{i_1, i_2, i_3, ..., i_n\}$. The event identifiers can be grouped by the case or process instance identifier (x) using the following subset definition $\iota_x : \{e \in \mathcal{E} | \iota = x\}$. Since the exact structure of a specific event log is not known in advance, this contribution will use a generic and *flexible definition* of the function that is used to query for events. For specific business rules the function notation will contain all the relevant event properties for that function, e.g. for activity order rules it should suffice to provide the activity (a_i) and event type (t_j) this results in the following notation $\alpha(a_i, t_j)$. However, when a specific role (r_k) is required the function notation will additionally include at least this role, consequently $\alpha(a_i, t_j, r_k)$ etc.

4.3.2. Specifying Dynamic Process Aspects in Linear Temporal Logic

Since business process management systems can be considered as reactive systems, patterns can be modeled using linear temporal logic (LTL). This results in formulae that can be interpreted over linear state sequences, which allows to define internal controls that can be interpreted over the entire set of events (i.e. the different states) of a single process instance. However, there exists a crucial difference between a business process instance and a reactive system, which is the trace length. Regular reactive systems are considered non-terminating, which is reflected in the infinite semantics of regular LTL. Therefore a bounded version of the LTL definition (in accordance with [22]) should be used. Secondly, whereas each element of a trace in regular LTL may consist of a set of properties, an audit trail contains exactly one event for each element [46].

Definition 4: LTL formula. An LTL formula p over a subset of $\mathcal{E}$ is a function $p : \mathcal{E}^* \rightarrow \{true, false\}$, with $\sigma \vDash p$ denoting that the formula p satisfies trace σ (i.e. $p(\sigma) = true$) and $\sigma \nvDash p$ denoting that formula p does not satisfy trace σ (i.e. $p(\sigma) = false$). For all LTL formulas p and q *true, false,* $\neg p$, $p \wedge q$, $p \vee q$, $\Box p$, $\Diamond p$, $\bigcirc p$, p *U* q *and* p *W* q are LTL formulas as well. The semantics and syntax (in the Manna/Pnueli notation) of LTL are defined as follows:

- Atomic proposition: **proposition:** $\sigma \vDash p$ if and only if $p = \sigma[1]$
- Boolean connectives: **not** ($\neg$)**:** $\sigma \vDash \neg p$ if and only if $\sigma \nvDash p$, **and** ($\wedge$)**:** $\sigma \vDash p \wedge q$ if and only if $\sigma \vDash p$ and $\sigma \vDash q$, **or** ($\vee$)**:** $\sigma \vDash p \vee q$ if and only if $\sigma \vDash p$ or $\sigma \vDash q$, **implication** ($\Rightarrow$)**:** $\sigma \vDash p \Rightarrow q$ if and only if $\sigma \vDash \neg p \vee q$, **equivalence** ($\Leftrightarrow$)**:** $\sigma \vDash p \Leftrightarrow q$ if and only if $\sigma \vDash (p \wedge q) \vee (\neg p \wedge \neg q)$, **true** (true)**:** $\sigma \vDash true$ if and only if $\sigma \vDash p \vee \neg p$ and **false** (false)**:** $\sigma \vDash false$ if and only if $\sigma \nvDash true$
- Temporal connectives: **next** ($\bigcirc$)**:** $\sigma \vDash \bigcirc p$ if and only if $\sigma^{2\rightarrowtail} \vDash p$, **until** (U)**:** $\sigma \vDash pUq$ if and only if $(\exists_{1 \leq i \leq n} : (\sigma^{i\rightarrowtail} \vDash q \wedge (\forall_{1 \leq j < i} : \sigma^{j\rightarrowtail} \vDash p))$, **eventually** ($\Diamond$)**:** $\sigma \vDash \Diamond p$ if and only if $\sigma \vDash trueUp$, **always** ($\Box$)**:** $\sigma \vDash \Box p$ if and only if $\sigma \vDash \neg\Diamond\neg p$ and **weak until** (W)**:** $\sigma \vDash pWq$ if and only if $\sigma \vDash (pUq) \vee (\Box p)$

The main advantage LTL has to offer in the context of business process compliance checking, is the ability to express relative time properties between states. This becomes especially clear in controls that specify that something has to hold eventually or that something has to hold until. Moreover, LTL has been successfully used in model checking (e.g. [7]), in ProM plug-ins (e.g. [65]) and declarative process modeling (e.g. [66, 48]). Examples include (written using the Manna/Pnueli notation):

- Activity inclusion rule: Activity a_1 and activity a_2 are mutually inclusive (must be tested for each process instance x)
  i.e. $\Diamond\alpha(x, a_1, t_c) \Leftrightarrow \Diamond\alpha(x, a_2, t_c)$
- Activity start precondition: Activity a_1 can only be executed in a process instance if expression μ holds (must be tested for each process instance x)
  i.e. $(\neg\alpha(x, a_1, t_s))W\mu$

4.3.3. Specifying Static Process Aspects in First Order Logic

In the context of static property rules there is only a need to evaluate one specific process state, i.e. one specific event. Since the time aspect is not crucial here, the semantics of the first order logic expression language should suffice. Examples are:

- Prohibited role-based allocation rule: Activity a_1 must not be performed by an originator of role r_1
  $\neg(\exists i \in \iota : \alpha(i, a_1, t_c, o, r_1))$
- Absolute time rule: Activity a_1 must be performed before T_0 (with timestamp = τ)
  $\neg(\exists i \in \iota : (\alpha(i, a_1, t_c, \tau) \wedge (\tau > T_0)))$

4.3.4. Composed Business Rule Patterns

In reality there will be a need for more sophisticated rule patterns than the atomic ones presented in the rule pattern taxonomy. Most of the desired internal control (effectiveness test) patterns can be obtained through a **composition of multiple atomic rule patterns**. These compositions can be obtained through the use of logical connectors such as $\vee, \wedge, \rightarrow, \neg, etc.$ Two types of combination can be identified:

- *Combination of rule patterns of different types*
  For example a combination of a simple response and precedence rule: A *register*

insurance policy activity (a_1) should be followed by a *process premium payment* activity (a_2) and a *process premium payment* activity should be preceded by a *register insurance policy* (must be tested for each process instance x)
i.e. $\Box(\alpha(x, a_1, t_c) \Rightarrow \Diamond\alpha(x, a_2, t_c)) \wedge (\neg(\alpha(x, a_2, t_s) \vee \alpha(x, a_2, t_c)))W\alpha(x, a_1, t_c)$

- *Combination of rule patterns of the same type*
  For example a combination of two precedence constraints: A *pay for damages* activity (a_1) must be preceded by a *evaluate claim* activity (a_2) or an *provide expert review* activity (a_3) (must be tested for each process instance x)
  i.e. $(\neg(\alpha(x, a_1, t_s) \vee \alpha(x, a_1, t_c)))W(\alpha(x, a_2, t_c) \vee \alpha(x, a_3, t_c))$

4.4. Determining the Rule-Scope for Configured Rule Patterns

Defining the scope of a business rule or control can be considered as specifying the applicability range of that business rule in terms of a certain dimension. The advanced rule-based approach for process mining seems to imply the **business process dimension**, for which we distinguish three levels global, (set of) business processes and (set of) business process instances.

The *global process scope* requires that each process in an organizations business operations complies with that rule. Typically a 'record financial transaction' activity must be performed by an agent with role accountant. Secondly, business rules may be imposed to a *specific (set of) business processes*. This might be the case for a claim handling process where the significant settlement proposals (e.g. above $10000) must be approved by a manager. An insurance company might have different insurance products different claim handling procedures, but it is reasonable to assume that for each process a management approval will be needed for a significant settlement proposal. Thirdly, the business rule scope may be restricted to *a (set of) business process instances* with a specific characteristic. Within the context of an order-to-cash a business rule may be defined that non-recurring customers give an advance before their order can be processed.

While the process dimension seems natural in the rule-based process mining setting, other dimensions could be valuable. Typical examples are the stakeholder dimension (e.g. customers, employees, management) or the department/devision dimension (e.g. ranging from a specific department to multiple divisions). Additionally, for some business rules a multidimensional scope definition may be required.

4.5. Extending the Rule-Based Compliance Checking Approach: Process Reconciliations & Document Driven Information Systems

This section briefly discusses two interesting extensions for the proposed comprehensive rule-based compliance checking approach with process mining: the duality between process activities and process artifacts and the relationship between process elements of different processes which enable interprocess reconciliations.

4.5.1. Duality Between Activities and Artifacts: Compliance Checking for Document Driven Information Systems

The proposed rule patterns are all specified in terms of activities. However, in addition to process-aware information systems there is a significant amount of contemporary organizations that use a *document-driven (i.e. artifact-centered) information system*. An activity-artifact coexistence business rule that specifies that a specific artifact must

exist before an activity can be completed, implicitly defines a **duality between process activities and process artifacts**. Since process activities and process artifacts are interchangeable, the rule patterns can be easily adapted and used in a data-driven environment. For example, testing whether a provide expert review activity has been performed before a pay for damages activity has been executed in a single process instance, is equal to testing whether an expert report was created before a payment record was recorded for a single business case.

4.5.2. Interprocess Reconciliations

A final remark deals with the origin of the events used for the compliance analysis. Until now process mining techniques always assumed that the relevant events were contained in one event log. However, contemporary organizations use a multitude of information systems which each produce a multitude of event logs for different processes. There may exist crucial **relationships between process elements of different processes** that are worthwhile to analyze for anyone in a control function. While the interpretation of the rule patterns can be easily extended to include cross-process relations, the implementation between them remains far from trivial in cases where there is no exact mapping between case identifiers in multiple systems. Using data analysis techniques to interpret and compare event data might enable the creation of such a mapping.

5. Populating the Taxonomy: Identifying Relevant Rule Patterns for Compliance Checking & Risk Management

Specific sets of business rule types will be defined for each possible combination of elements from both the process mining perspective dimension and the rule restriction focus dimension. When selecting the business rule types we took into account the process mining technical feasibility and the auditability of each business rule type. Hence it appears that important (detective) controls, such as direct supervision, could not be included in the taxonomy as they are not technically feasible and/or not auditable. Each of the following subsections introduce the different identified business rule types for a specific process mining perspective, within the subsections an ordering according to rule restriction focus is maintained.

5.1. Functional Perspective

The functional perspective on business processes deals with the occurrence of process activities (and related artifacts) in a process instance. In the business rule class where the focus is put on **cardinality**, we typically observe activity cardinality rule subtypes. These business rule subtypes represent the rules that implement a restriction on the number of occurrences of a certain activity in a specific process instance [23].

Combining both the functional perspective and a **coexistence** focus, results in business rule types that describe the possibilities of *coexistence between activities of different activity types* and of *coexistence between certain activity and event types*. The subtypes range from required coexistence to a forced non-coexistence between activity types and activity-event type combinations [48]. A **dynamic data-driven** restriction focus results in rule types that link the existence or absence of an activity of a certain type to a data-oriented condition. This data-oriented condition can be true from the beginning

(e.g. if based on case data) or can become true during the process instance execution (e.g. specific event data). **Relative time rules** in the context of the functional process perspective can refer to the rules that link the existence or absence of a certain activity in a process instance to a relative time condition, such as before X time units starting from the process start.

In addition to rule types that appeal to the dynamic aspect of a process, **static property rule** types can also be distinguished. The *event-artifact coexistence rule* subtypes deal with the relation between activities and certain process related artifacts in a particular process state. This rule type is related to the postcondition as specified in the Web Service Modeling Ontology (WSMO) [49].

5.2. Control-Flow Perspective

The control-flow perspective on business processes focuses on the ordering of the process elements in a process instance. **Coexistence rules** within this process perspective deal with specifying ordering rules between activities of different activity types. Whereas the *non-overlapping activity rule* subtype can be used for just avoiding the concurrent execution of specific activities, the *activity order rule* subtypes define exact ordering relationships between activities of certain activity types. A lot of research effort towards defining activity order rules has been performed by Pesic et al., e.g. [48].

The **data-driven rule** types consist of business rules that define data conditions which need to be satisfied prior to the start of an activity of a particular activity type, i.e. *data-driven activity preconditions*. These rule subtypes can be related to the preconditions as specified in the WSMO [49].

A combination of **relative time rules** and the control-flow perspective results in two subtypes of time-driven control-flow rules: the activity time rule subtype and the inter-activity time rule subtype. Business rules that belong to the *activity time rule* subtype specify a condition on the allowable execution time of an activity of a specific activity type (e.g. minimum or maximum duration). In contrast business rules of the *inter-activity time rule* subtype define comparable conditions on the allowable time interval between activities of (different) activity types. Based on the timestamp an *absolute time rule type* can be specified for the **static property** focus.

There is, however, no **cardinality-based** rule type specified for the control-flow perspective. The main reason for this is the use of varying granularity: activities in coarse grained business processes may be composed of subprocesses in finer grained representations of the same business process. Therefore, one could also use the activity cardinality rules in this context.

5.3. Organizational Perspective

The organizational perspective on business processes deals with the (human) resources aspect of the business processes, i.e. who performs which activity in the process. When focusing on **cardinality**, the *originator cardinality rule* type was identified. This rule type allows to put restrictions on the allowable number of executions of activities of an activity type by a specific agent within the context of a single process instance [61]. The business rule class that is defined by the combination of the organizational perspective with a **coexistence** focus defines three main business rule types: the segregation of duties rule type, the binding of duties type and the temporal engagement rule type. Whereas

	Cardinality-Based Rules	Coexistence Rules	Dynamic Data-Driven Rules	Relative Time Rules	Static Property Rules
Functional Perspective	**Activity Cardinality Rule** *Activity existence rule* *Activity absence rule* *Activity range rule*	**Activity Coexistence Rule** *Activity inclusion rule* *Activity substitution rule* *Responded existence rule* *Activity choice rule* **Event-Activity Coexistence Rule** *Event-activity inclusion rule* *Event-activity exclusion rule* *Event-activity choice rule*	**Data-Driven Existence Rule** *Data-driven activity inclusion rule* *Data-driven activity exclusion rule*	**Time-Oriented Activity Existence Rule** *Timed activity existence rule* *Timed activity absence rule*	**Event-Artifact Coexistence Rule** *Activity-artifact coexistence rule* *Non-activity event-artifact coexistence rule*
Control Flow Perspective		**Non-Overlapping Activity Rule** **Activity Order Rule** *Simple response rule* *Simple precedence rule* *Alternate response rule* *Alternate precedence rule* *Chain response rule* *Chain precedence rule*	**Data-Driven Activity Precondition rule** *Activity start precondition rule* *Activity complete precondition rule*	**Time-Driven Control-Flow Rule** *Activity time rule* *Inter-activity time rule*	**Absolute Time Rule**

Table 1: Advanced Rule-Based Controls Classification

	Cardinality-Based Rules	Coexistence Rules	Dynamic Data-Driven Rules	Relative Time Rules	Static Property Rules
Organization Perspective	**Originator Cardinality Rule**	**Segregation of Duties** *Conflicting roles* *Dynamic segregation of duties* *Operational segregation of duties* *History-based segregation of duties* **Binding of Duties** **Temporal Engagement Rule**	**Exogenous Authorization Rule** **Originator Attribute Rule** *Static originator attribute rule* *Dynamic originator attribute rule*	**Temporal Deontic Rule** *Temporal obligation rule* *Temporal permission rule* *Temporal prohibition rule*	**Static Authorization Rule** *Role-based activity authorization rule* *Prohibited role-based allocation rule* *Not-optimal allocation rule* **Required Originator Attribute Rule** **Delegation Rule** *Prohibited role acquisition rule* *Delegation authorization rule* *Retract restriction rule*
Data Perspective	**Event Data Cardinality Rule** *Mandatory event data rule* *Event data multiplicity constraint*	**Event Data Coexistence Rule** *Disjunctive event data rule* *Mutually exclusive event data rule*	**Derived Event Data Rule** *Arithmetic derivation rule* *Classification rule* **Event Data Comparison Rule** *Event data equality rule* *Event data exclusion rule*	**Dynamic Integrity Rule** *Time-oriented integrity rule* *Activity-oriented integrity rule* *Event-oriented integrity rule*	**Event Data Value Rule** *Event data value set rule* *Event data value range rule* *Event data uniqueness rule* *Irreflexive event data rule* **Event Data Format Rule**

Table 2: Advanced Rule-Based Controls Classification (Continued)

the *segregation of duties* rule subtypes focus on avoiding the risks related to letting the same agent perform all activities, the *binding of duties* rule type requires that a set of activities is performed by the same person. These rule subtypes have been extensively researched, e.g. in [19, 39]. The *temporal engagement rule* type makes it possible to verify if a specified person had a particular role/function at a certain point in time.

Business rule types with a **dynamic data-driven** focus enable the specification of organizational preconditions. *Exogenous authorization rules* deal with conditions and factors external to the process instance executions. *Originator attribute rules* define restrictions on who can perform an activity based on a combination of event/case data and originator specific data. The **relative time rules**' main rule subtypes are linked to *temporal deontic rules*. These rules are based on the concept of a deontic assignment that can represent amongst others the obligation or permission of an agent to perform a particular activity while respecting a time constraint, e.g. [80].

The **static property rule** types include the static authorization rule type, the required originator attribute rule type and the delegation rule type. The *static authorization subtypes* allow for the specification of non-evolving authorizations controls. Role-based authorization and prohibited allocation rules define respectively the right to perform an activity and the prohibition to perform an activity. Both are actively been researched for example in [19]. Suboptimal allocations that may harm efficiency and effectiveness can be traced down with non-optimal allocation rules. The second main type of static property rules, the *required originator attribute rules*, deals with verifying whether an agent possesses certain characteristics needed to perform the activity. Finally, the *delegation rule* subtypes cover the whole spectrum of concerns that are related to role delegation: from allowable allocations, over delegation authorizations to retract policies [76].

While the main organizational rules will focus on restricting the rights of specific individuals or individuals of a certain role, several rule types could be specified for different 'agents' (e.g. an organizational entity).

5.4. Data Perspective

The data or informational perspective on business processes represents the informational elements that are used, produced or manipulated during the process, as well as the relationships among them. A distinction is made between data elements that relate to events (i.e. event data, e.g. invoice amount in event related to a pay for damages activity) and data elements that are specified for a specific business process instance (i.e. case data, e.g. claim for car insurance or premium customer involved).

In the context of **cardinality-based rules**, the *event data cardinality rule* subtypes can be distinguished. Each of these subtypes deals with restricting the allowed number of instances of a certain data element type for a single instance of a specific event type. The **coexistence rules**, on the other hand, specify co-occurrence restrictions for data elements of different types within the context of a single instance of a specific event type.

The **dynamic data driven rule** types define the value of specific event data elements in terms of the value of other data elements. Whereas the *derived event data rule* subtypes provide some sort of expression to determine the value of a data element, the *event data comparison rule* subtypes define value comparison restrictions (e.g. is equal, is not equal, is larger than, etc.) based on the value of other data elements and possibly over multiple

events. These rules are related to the set comparison constraints in the Object-Role Modeling Approach, e.g. [31]. **Relative time rules** in the data perspective mainly focus on specifying the *dynamic integrity* of data elements value. Admissible changes in the data elements value are specified in terms of exact time, the execution of an activity [29] or the occurrence of an event.

Included in the **static property rule** types for the data perspective are both business rule types that specify *acceptable event data values* (in terms of sets, ranges, etc.) and that impose the *event data format*, see also [31].

6. Evaluating Opportunities and Challenges

This section discusses the opportunities for strategic advantages that can be obtained with rule-based compliance checking approaches for process mining over more traditional auditing and risk management techniques. In addition to the fact that the auditors credibility is generally higher when they use new technologies [59], we distinguish three opportunities: high issue detection effectiveness (pursuing absolute assurance), obtaining persuasive evidence and achieving complete auditor independence. These advantages all contribute to a more efficient and effective control environment. Furthermore, we discuss the challenges related to event log data quality & preprocessing, to distortions in interpretation & pattern design and to the implementation of continuous auditing/monitoring.

6.1. Opportunity 1: High Issue Detection Effectiveness - Approaching Absolute Assurance

The major enabler for process mining as a computer assisted auditing technique (CAAT) or as a risk event identification approach will be the increased level of assurance. Assurance is a measure used to indicate the level of certainty that an agent has obtained about his statements, related to the controls (e.g. the absence of approval issues). Different levels of assurance have been identified: limited, reasonable and absolute assurance [34]. However, no strict definitions have been provided in the literature.

Process mining techniques enable the agent to perform the tests on the *full* population and thereby offering (near) **absolute assurance**, with a marginally higher cost in terms of processor time compared to sample-based testing with process mining.

6.2. Opportunity 2: Obtaining Persuasive Evidence

In addition to the high issue detection effectiveness, the persuasiveness of the evidence obtained through process mining techniques is expected to be high. This persuasiveness is strongly related to the competence and the sufficiency of the evidence that can be attained. According to [2], the **competence of the evidence** is mainly determined by the independence of the provider, the evaluator's direct knowledge, the degree of objectivity and the timeliness (referring both to the period covered and the ability to reduce the time-delay). While most of these determinants can be easily related to the basic advantages of process mining as discussed in the previous sections, a continuous monitoring / auditing approach is required in order to outperform the traditional monitoring / auditing techniques on time-delay related timeliness. Due to the facts that the whole population of transactions can be efficiently inspected and that the risk of overlooking an issue is extremely low, the **sufficiency of the evidence** will be optimal.

6.3. Opportunity 3: Realizing Complete Auditor Independence

The value of an audit (report) is largely dependent on the independence of the persons involved in the auditing process from the organization that is being audited [2]. Two major architecture types have been proposed in the context of analytic CAATs; internal (e.g. embedded auditing modules architecture [27]) and **external** (e.g. monitoring and control architecture [75] and process mining based auditing) auditing modules. In contrast to the internal auditing module architectures that directly affect the clients information systems, the external auditing module architectures only use the output of the clients software. This gives the latter a major advantage from the auditor independence point of view.

Firstly, the functionality of the audit modules is **designed and maintained by the auditor** (or his/her firm), without any possible interference from the client or his system administrators. Secondly, architectures based on external auditing modules can fully guarantee a **non-existence of a-priori knowledge** with the client about the auditing approach and procedures. In contrast, the implementation of internal auditing modules would require cooperation of the client, which might result in a-priori knowledge with the client. Consequently, the client could use this information to conduct fraudulent behavior that would not trigger any alarms with the auditor. Other advantages of external over internal auditing modules include: the absence of legal liability for side-effects (e.g. decrease in performance) [40] of the modules in the client's information system and the reusability of the modules as they are not written in information system specific languages (e.g. ERP programming languages) [17].

6.4. Challenge 1: Event Log Data Quality & Preprocessing

Compliance and risk analyses require event logs that meet high quality standards in order to derive meaningful conclusions. Four main quality criteria can be distinguished: the recording of the events should be done in a **trustworthy, securely, systematic and accurate** manner. The data in an event-log of an information system must be recorded in a *trustworthy* manner, e.g. ID fraud or backdating must be impossible. Additionally, these recordings must be kept *securely*, which refers to the prevention of any tampering with the data after the recording. The third quality criteria, *systematic recording*, reflects the need for a timely recording of every important business event (i.e. completeness). In addition to start and end events for activities, the recording of denial events (resulting from trying to perform unauthorized activities) and cancel events might be interesting from a control/audit perspective as they might indicate potential fraud attempts. *Accuracy* can be regarded as a measure that reflects the correctness of the representation in the event-log compared to a real-life event. This can be determined both on syntactical as semantical level. In [36], for example, it is argued that major ERP systems can contain missing and faulty data values. While trustworthy & securely recording looks at the possibilities for humans to influence the recorded data, accuracy focuses on the ability of the provenance system to correctly record the data (i.e. noise because of technical errors).

The actual preprocessing challenges are: **the selection of a suitable instance level, dealing with convergence/divergence and attribute selection**. Instance level selection is partially determined by the scope of the compliance or risk analysis (e.g. audit), i.e. the business object under evaluation. However, the major determinant will

be the granularity of the available events (e.g. availability of order related events versus order line related events). Related to the instance selection challenge are convergence and divergence issues [57]. Convergence refers to the phenomenon in which the same activity is executed on multiple process instances at once, e.g. a client pays for multiple orders at once. Divergence on the other hand refers to the business situation in which the same activity is executed multiple times for one process instance, e.g. for each order line of a single order a delivery activity is recorded. Another important preprocessing decision is the attribute selection. When too many attributes are included in the event log, the log becomes unnecessarily large, hard to handle, difficult to load in various tools, etc. On the contrary, if too few attributes are included some analyses will become impossible. The scope of the analysis should help clear this up.

6.5. Challenge 2: Distortions in Interpretation and Pattern Design

Comparable to other compliance and risk analysis approaches interpretation and design distortions might occur. Legislation, policies and directives can remain vague and ambiguous, therefore **assumptions** are often made in the interpretation phase [41]. During the actual configuration or design of the specific set of business rules, the analysts might be confronted with **expressibility limitations** for a small number of extremely rare and context specific internal controls. Additionally, the **completeness assumption** of an event log, every possible behavior is covered in the event log, can often be challenged. This is especially important for the interpretation of risk assessment analysis results.

Compared to the other process mining approaches to risk and compliance analysis, however, the number of possible distortions remains limited. Both conformance checking and delta analysis are confronted with process overspecification, implicit rules that are not specified by any directive are included in the process model. Visualization and delta analysis techniques additionally face the trade-off between specificity and generalization of behavior that is made by process mining algorithms.

6.6. Challenge 3: Implementing Rule-Based Continuous Auditing/Monitoring

Traditionally, an important time-delay can be observed between the occurrence of important business events and the monitoring/audit report due to the **periodic nature** of most monitoring/auditing models. In the process mining based management/audit models this periodic nature and the related time-delay can also be observed. As a result of this delay, the **information contained in the management/audit report might become less useful or beneficial** for its user and therefore significantly affects his/her ability to make well-grounded business decisions [4].

Both continuous monitoring (implemented by management) and continuous auditing focus on reducing this time-delay by providing reporting/assurance **simultaneously with, or a short period of time after**, the occurrence of events related to the subject under investigation [8]. Consequently continuous monitoring and auditing systems may unlock interesting opportunities; e.g. timely meeting regulatory requirements, promptly identifying irregularities or enhancing a stakeholder's ability to make decisions.

Developing continuous auditing/monitoring models based on process mining can be an interesting challenge. The architectural methodology that could be used for implementing continuous auditing with process mining techniques is based on a **monitoring control**

layer (MCL) [75]. This monitoring control layer would consist of adapted process mining techniques, which are using the event logs/event streams of the information system as an input [72].

7. Conclusion & Outlook

Process mining research has been characterized by a narrow research focus on theoretical improvements and especially by the advancement of process discovery techniques. Therefore only a partial fit between, on the one hand, compliance checking & risk management and, on the other hand, process mining techniques did exist. Aspects of this rather limited fit include: the ignorance of case and event data, the reasonable doubt about the correctness of designed process models including overspecification (for conformance checking & delta analysis), the need for balance between precision and generality potentially whipping out suspicious behavior (for process discovery), etc.

In this research report we proposed a comprehensive rule-based compliance checking and risk management approach as a possible solution to eliminate the limited fit. The approach enables analysts to uncover compliance failures as well as to identify and assess potential risks. Improvements can be found in the ability to take additional data into account, the reduction of possible distortions (including over specification), the ability to deal with noise (no need for generalization), etc. Additionally, a rule-based compliance checking and risk management approach provides information on a potential compliance risk, whereas recall/precision metrics only provide a process-wide indicator. This contribution proposed a extensive content-based business rule classification, resulting in an extensive set of rule patterns that is fit to be used in a common business setting. Whereas the comprehensibility of the rule patterns is high due to the use of native English, the formal grounding removes every room for interpretation. Finally, an evaluation containing the major opportunities (i.e. effectiveness, persuasive evidence and audit independence) and challenges (i.e. data quality & preprocessing, distortions in interpretation & pattern design and continuous monitoring/auditing) was presented. A logical future step in our research is to further test this approach on real-life cases and to tackle the identified challenges. Focus will be placed on the development of a continuous monitoring/auditing approach based on process mining techniques.

Acknowledgements

We would like to thank the K.U.Leuven research council for financial support under grant OT/10/010: Business Process Mining: New Techniques and Evaluation Metrics and the Flemish research council for financial support under the Odysseus grant B.0915.09.

References

[1] R. Agrawal, C. Johnson, J. Kiernan, and F. Leymann. Taming compliance with sarbanes-oxley internal controls using database technology. In *Proceedings of the 22nd International Conference on Data Engineering*, pages 92–92. IEEE, 2006.

[2] A.A. Arens, R.J. Elder, and M.S. Beasley. *Auditing and assurance services: An integrated approach.* Pearson Education, 2005.

[3] J. Bang-Jensen and G.Z. Gutin. *Digraphs: theory, algorithms and applications*. Springer Verlag, 2010.
[4] ISACA Standards Board. Continuous auditing: Is it fantasy or reality? *Information Systems Control Journal*, 5, 2002.
[5] F. Casati, S. Castano, M. Fugini, I. Mirbel, and B. Pernici. Using patterns to design rules in workflows. *IEEE Transactions on Software Engineering*, 26(8):760–785, 2000.
[6] F. Chesani, P. Mello, M. Montali, F. Riguzzi, M. Sebastianis, and S. Storari. Checking compliance of execution traces to business rules. In *Business Process Management Workshops*, pages 134–145. Springer, 2009.
[7] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model checking*. MIT Press, January 2000.
[8] Wood Committee. Research report: continuous auditing. Technical report, Canadian Institute of Chartered Accountants & American Institute of Certified Public Accountants, 1999.
[9] J.E. Cook and A.L. Wolf. Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology*, 7(3):215–249, 1998.
[10] J.E. Cook and A.L. Wolf. Software process validation: quantitatively measuring the correspondence of a process to a model. *ACM Transactions on Software Engineering and Methodology*, 8(2):147–176, 1999.
[11] COSO. Enterprise risk management - integrated framework. Technical report, Committee of Sponsoring Organizations of the Treadway Commission, 2004.
[12] B. Curtis, M.I. Kellner, and J. Over. Process modeling. *Communications of the ACM*, 35(9):75–90, 1992.
[13] C.J. Date. *What not how: the business rules approach to application development*. Addison-Wesley Professional, 2000.
[14] H. de Beer. The LTL checker plugins: A reference manual. Technical report, Eindhoven University of Technology, 2004.
[15] A.K.A. de Medeiros, W.M.P. van der Aalst, and C. Pedrinaci. Semantic process mining tools: core building blocks. In *16th European Conference on Information Systems*, pages 1953–1964. Citeseer, 2008.
[16] A.K.A. de Medeiros, B.F. van Dongen, W.M.P. van der Aalst, and A. Weijters. Process mining: extending the α-algorithm to mine short loops. Technical report, Eindhoven University of Technology, 2004.
[17] R.S. Debreceny, G.L. Gray, J.J.J. Ng, K.S.P. Lee, and W.F. Yau. Embedded audit modules in enterprise resource planning systems: implementation and functionality. *Journal of Information Systems*, 19:7, 2005.
[18] R. Dijkman, M. Dumas, B. Van Dongen, R. Kaarik, and J. Mendling. Similarity of business process models: Metrics and evaluation. *Information Systems*, 36(2):498–516, 2011.
[19] D. Ferraiolo, D.R. Kuhn, and R. Chandramouli. *Role-based access control*. Artech House Publishers, 2003.
[20] D.R. Ferreira. Applied sequence clustering techniques for process mining. *Handbook of Research on Business Process Modeling. IGI Global*, 2009.
[21] D. Follesdal and R. Hilpinen. *Deontic logic: An introduction*, chapter Deontic Logic: Introductory and Systematic Readings, pages 1–35. D. Reidel Publishing Company, Dordrecht, 1971.
[22] D. Giannakopoulou and K. Havelund. Automata-based verification of temporal properties on running programs. In *Proceedings of the 16th Annual Conference on Automated Software Engineering*, pages 412–416. Published by the IEEE Computer Society, 2001.
[23] S. Goedertier, R. Haesen, and J. Vanthienen. Rule-based business process modelling and enactment. *International Journal of Business Process Integration and Management*, 3(3):194–207, 2008.
[24] S. Goedertier, D. Martens, J. Vanthienen, and B. Baesens. Robust process discovery with artificial negative events. *The Journal of Machine Learning Research*, 10:1305–1340, 2009.
[25] R. Gopal, J.R. Marsden, and J. Vanthienen. Information mining-reflections on recent advancements and the road ahead in data, text, and media mining. *Decision Support Systems*, 2011.
[26] G. Governatori and Z. Milosevic. Dealing with contract violations: formalism and domain specific language. In *Proceedings of the International Enterprise Distributed Objet Computing Conference*, pages 47–57, 2005.
[27] S.M. Groomer and U.S. Murthy. Continuous auditing of database applications: An embedded audit module approach. *Journal of Information Systems*, 3(2):53–70, 1989.
[28] Business Rules Group. Defining business rules: What are they really?(3rd edition). Technical report, Business Rules Group, 2000.
[29] Object Management Group. Semantics of business vocabulary and rules (SBVR). Technical report,

Technical Report dtc/06–03–02, 2006.
[30] G. Guizzardi. Ontological foundations for structural conceptual models. 2005.
[31] T. Halpin. Object-role modeling (ORM/NIAM). *Handbook on Architectures of Information Systems*, pages 81–103, 2006.
[32] S.M. Huang, D.C. Yen, Y.C. Hung, Y.J. Zhou, and J.S. Hua. A business process gap detecting mechanism between information system process flow and internal control flow. *Decision Support Systems*, 47(4):436–454, 2009.
[33] S.Y. Hwang and W.S. Yang. On the discovery of process models from their instances* 1. *Decision Support Systems*, 34(1):41–57, 2002.
[34] IFAC. *Handbook of international auditing, assurance and ethics pronouncements*. International Federation of Accountants, 2008.
[35] IIA. Position statement: The role of internal audit in enterprise-wide risk management. Technical report, The Institute of Internal Auditors, 2004.
[36] J.E. Ingvaldsen and J.A. Gulla. Preprocessing support for large scale process mining of SAP transactions. In *Proceedings of the 2007 International Conference on Business Process Management*, pages 30–41. Springer, 2007.
[37] M. Jans, N. Lybaert, and K. Vanhoof. Internal fraud risk reduction: Results of a data mining case study. *International Journal of Accounting Information Systems*, 11(1):17–41, 2010.
[38] M. Jans, N. Lybaert, K. Vanhoof, and J.M. van der Werf. A business process mining application for internal transaction fraud mitigation. *Expert Systems with Applications*, 38(10):13351–13359, 2011.
[39] K. Knorr and H. Stormer. Modeling and analyzing separation of duties in workflow environments. *Trusted Information*, pages 199–212, 2002.
[40] J.R. Kuhn Jr and S.G. Sutton. Continuous auditing in ERP system environments: The current state and future directions. *Journal of Information Systems*, 24(1).
[41] K.J. Lee and B. Jeon. Analysis of best practice policy and benchmarking behavior for government knowledge management. *Knowledge Management in Electronic Government*, pages 70–79, 2004.
[42] Y. Liu, S. Muller, and K. Xu. A static compliance-checking framework for business process models. *IBM Systems Journal*, 46(2):335–361, 2007.
[43] R. Lu, S. Sadiq, and G. Governatori. Compliance aware business process design. In *Business Process Management Workshops*, pages 120–131. Springer, 2008.
[44] R.S. Mans, M.H. Schonenberg, M. Song, W.M.P. Aalst, and P.J.M. Bakker. Application of process mining in healthcare–a case study in a dutch hospital. *Biomedical Engineering Systems and Technologies*, 25:425–438.
[45] M. Montali. Declarative process pining. *Specification and Verification of Declarative Open Interaction Models*, pages 343–365, 2010.
[46] M. Pesic. *Constrained-based workflow management systems: Shifting control to users*. PhD thesis, Eindhoven University of Technology, 2008.
[47] M. Pesic, M.H. Schonenberg, N. Sidorova, and W.M.P. Van Der Aalst. Constraint-based workflow models: Change made easy. In *Proceedings of the 2007 OTM Confederated international conference on the move to meaningful internet systems*, pages 77–94. Springer-Verlag, 2007.
[48] M. Pesic and W.M.P. van der Aalst. A declarative approach for flexible business processes management. In *Business Process Management Workshops*, pages 169–180. Springer, 2006.
[49] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web service modeling ontology. *Applied Ontology*, 1(1):77–106, 2005.
[50] R. Ross. *Business Rule Conepts (3rd Edition)*. Business Rule Solutions, 2009.
[51] R.G. Ross. *Principles of the business rule approach*. Addison-Wesley Professional, 2003.
[52] A. Rozinat and W.M.P. van der Aalst. Conformance testing: Measuring the fit and appropriateness of event logs and process models. In *Business Process Management Workshops*, pages 163–176. Springer, 2006.
[53] A. Rozinat and W.M.P. van der Aalst. Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1):64–95, 2008.
[54] V. Rubin, C. Gunther, W.M.P. van der Aalst, E. Kindler, B. van Dongen, and W. Schafer. Process mining framework for software processes. *Software Process Dynamics and Agility*, pages 169–181, 2007.
[55] S. Sadiq, G. Governatori, and K. Namiri. Modeling control objectives for business process compliance. *Business Process Management*, pages 149–164, 2007.
[56] S.W. Sadiq, M.E. Orlowska, and W. Sadiq. Specification and validation of process constraints for flexible workflows. *Information Systems*, 30(5):349–378, 2005.

[57] I.E.A. Segers. Investigating the application of process mining for auditing purposes. Master's thesis, Technische Universiteit Eindhoven, 2007.
[58] M. Song, C.W. Gunther, and W.M.P. Aalst. Trace clustering in process mining. In *Business Process Management Workshops*, pages 109–120. Springer, 2009.
[59] S. Sutton, R. Young, and P. Mckenzie. An analysis of potential legal liability incurred through audit expert systems. *Intelligent Systems in Accounting, Finance and Management*, 4:191–204, 1994.
[60] N. Syed Abdullah, S. Sadiq, and M. Indulska. Emerging challenges in information systems research for regulatory compliance management. In *Advanced Information Systems Engineering*, volume 6051/2010.
[61] K. Tan, J. Crampton, and C.A. Gunter. The consistency of task-based authorization constraints in workflow. In *Proceedings of the 17th IEEE Computer Security Foundations Workshop*, pages 155–169. IEEE, 2004.
[62] W.M.P. van Aalst, K.M. Van Hee, J.M. van Werf, and M. Verdonk. Auditing 2.0: Using process mining to support tomorrow's auditor. *Computer*, 43(3):90–93, 2010.
[63] W. Van der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.
[64] W.M.P. van der Aalst. Business alignment: Using process mining as a tool for delta analysis and conformance testing. *Requirements Engineering Journal*, 10(3):198–211, 2005.
[65] W.M.P. van der Aalst, H.T. De Beer, and B.F. van Dongen. Process mining and verification of properties: An approach based on temporal logic. *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, 3760/2005:130–147.
[66] W.M.P. Van Der Aalst and M. Pesic. DecSerFlow: Towards a truly declarative service flow language. *Web Services and Formal Methods*, pages 1–23, 2006.
[67] W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business process mining: An industrial application. *Information Systems*, 32(5):713–732, 2007.
[68] W.M.P. Van der Aalst and M. Song. Mining social networks: Uncovering interaction patterns in business processes. *Business Process Management*, pages 244–260, 2004.
[69] W.M.P. van der Aalst and B. Van Dongen. Discovering workflow performance models from timed logs. *Engineering and Deployment of Cooperative Information Systems*, 2480/2002:107–110.
[70] W.M.P. van der Aalst, B. van Dongen, C. Gunther, R. Mans, A. de Medeiros, A. Rozinat, V. Rubin, M. Song, H. Verbeek, and A. Weijters. ProM 4.0: Comprehensive support for real process analysis. *Petri Nets and Other Models of Concurrency–ICATPN 2007*, pages 484–494, 2007.
[71] W.M.P. Van der Aalst, B.F. Van Dongen, C. Gunther, A. Rozinat, H.M.W. Verbeek, and A. Weijters. ProM: the process mining toolkit. In *Proceedings of the International Conference on Business Process Management*, pages 1–4, 2009.
[72] W.M.P. van der Aalst, K. van Hee, J.M. van der Werf, A. Kumar, and M. Verdonk. Conceptual model for on line auditing. *Decision Support Systems*, 50, 2010.
[73] W.M.P. van der Aalst and A. Weijters. Process mining: a research agenda. *Computers in Industry*, 53(3):231–244, 2004.
[74] M. van Giessel and M.H. Jansen-Vullers. Process mining in sap r/3. *Master's thesis, Eindhoven University of Technology, Eindhoven*, 2004.
[75] M.A. Vasarhelyi, M.G. Alles, and A. Kogan. Principles of analytic monitoring for continuous assurance. *Journal of Emerging Technologies in Accounting*, 1(1):1–21, 2004.
[76] K. Venter and M. Olivier. The delegation authorization model: A model for the dynamic delegation of authorization rights in a secure workflow management system. *Proceedings of Information Security South Africa (ISSA02)*, 2002.
[77] G. Wagner. Rule modeling and markup. *Reasoning Web*, pages 251–274, 2005.
[78] A. Weijters and W.M.P. van der Aalst. Process mining: discovering workflow models from event-based data. In *Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001)*, pages 283–290. Citeseer, 2001.
[79] A. Weijters, W.M.P. van der Aalst, and A.K.A. de Medeiros. Process mining with the heuristics miner-algorithm. *Technische Universiteit Eindhoven, Tech. Rep. WP*, 166, 2006.
[80] P. Yolum and M.P. Singh. Reasoning about commitments in the event calculus: An approach for specifying and executing protocols. *Annals of Mathematics and Artificial Intelligence*, 42(1):227–253, 2004.
[81] M. zur Muehlen and M. Rosemann. Integrating risks in business process models. In *Proceedings of 16th Australasian Conference on Information Systems*, 2005.

FACULTY OF ECONOMICS AND BUSINESS
Naamsestraat 69 bus 3500
3000 LEUVEN, BELGIË
tel. + 32 16 32 66 12
fax + 32 16 32 67 91
info@econ.kuleuven.be
www.econ.kuleuven.be