

On the Distinction between Truthful, Invisible, False and Unobserved Events An Event Existence Classification Framework and the Impact on Business Process Analytics Related Research Areas

Filip Caron

Leuven, Leuven, Belgium., filip.caron@econ.kuleuven.be

Seppe vanden Broucke

Leuven, Leuven, Belgium., seppe.vandenbroucke@econ.kuleuven.be

Jan Vanthienen

Leuven, Leuven, Belgium., Jan.Vanthienen@econ.kuleuven.be

Bart Baesens

Leuven, Leuven, Belgium., bart.baesens@econ.kuleuven.be

Follow this and additional works at: <http://aisel.aisnet.org/amcis2012>

Recommended Citation

Caron, Filip; vanden Broucke, Seppe; Vanthienen, Jan; and Baesens, Bart, "On the Distinction between Truthful, Invisible, False and Unobserved Events An Event Existence Classification Framework and the Impact on Business Process Analytics Related Research Areas" (2012). *AMCIS 2012 Proceedings*. 24.

<http://aisel.aisnet.org/amcis2012/proceedings/DecisionSupport/24>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2012 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

On the Distinction between Truthful, Invisible, False and Unobserved Events

An Event Existence Classification Framework and the Impact on Business Process Analytics Related Research Areas

Filip Caron

Department of Decision Sciences and Information Management, Faculty of Business and Economics, KU Leuven

Filip.Caron@econ.kuleuven.be

Seppe vanden Broucke

Department of Decision Sciences and Information Management, Faculty of Business and Economics, KU Leuven

Seppe.vandenBroucke@econ.kuleuven.be

Jan Vanthienen

Department of Decision Sciences and Information Management, Faculty of Business and Economics, KU Leuven

Jan.Vanthienen@econ.kuleuven.be

Bart Baesens

Department of Decision Sciences and Information Management, Faculty of Business and Economics, KU Leuven

Bart.Baesens@econ.kuleuven.be

ABSTRACT

In this paper we present an event existence classification framework based on five business criteria. As a result we are able to distinguish thirteen event types distributed over four categories, i.e. truthful, invisible, false and unobserved events. Currently, several of these event types are not commonly dealt with in business process analytics research. Based on the proposed framework we situate the different business process analytics research areas and indicate the potential issues for each field. A real world case will be elaborated to demonstrate the relevance of the event classification framework.

Keywords

Information Systems Auditing, Event-Based Information Systems, Process Mining, Compliance Checking, Artificial Event Generation.

INTRODUCTION

For many organizations business process management has become crucial in effectively supporting their operations and consequently in achieving their goals (Buhl, Röglinger, Stöckl and Braunwrath, 2011). Process-aware information systems, implemented in the context of this business process management, often generate an immense amount of data describing the real process dynamics. While these systems provide a vast amount of analysis and visualization tools to monitor key performance indicators, an abundance of data about the process reality still remains untapped and concealed in so called event logs (van der Aalst and Weijters, 2004).

Recently, the business process analytics research area, including business process mining, has been gaining traction (zur Mühlen and Shapiro, 2010). This research field aims to provide solid tools and methods to deeply and thoroughly inspect, analyze and improve business processes, and thus providing answers on how useful insights can be derived from large business process event logs, without becoming overwhelmed by the amount of data which is available in these repositories (van der Aalst, Reijers, Weijters, van Dongen, de Medeiros, Song and Verbeek, 2007). At the same time, process mining analysis tasks strive to go beyond the simple one-dimensional reporting based techniques typically implemented in the process aware information systems.

Although process analytics thus enables decision makers and business analysts to extract valuable knowledge, there still exist some potentially problematic intricacies, which can present themselves when dealing with event logs. More specifically, we argue that common process analytics techniques usually deal with one specific kind (or type) of business events only: these

which are recorded in the event log under consideration and which (hopefully) correspond to some real-life activity. In this paper, we propose a complete ontology in order to distinguish between all the various sorts of business events that are important in the context of business process analytics. Note however that the proposed event existence classification framework is applicable in other (business) event related contexts as well, such as complex event processing (CEP), for instance.

This paper mainly contributes by providing an event existence classification framework deliberately based on business criteria, rather than technical aspects. The aim of this classification framework is threefold: firstly to raise awareness about the various types of events that can exist in a business context. Consequently removing the assumption that only the events that can be retrieved from event logs can or should be taken into account. Secondly, the framework provides a clear and unambiguous naming scheme that could result in a more effective communication in the business process analytics research. Finally, the framework enables a better orientation of the different business process analytics research areas and the ability to indicate potential issues in the research fields. Comprehensive examples of the different event types will be provided.

EXPLORATION OF THE EVENT EXISTENCE CLASSIFICATION CRITERIA

Previous research in event-based information systems and process analysis, e.g. process mining, has mainly focused on the behavior described by the registered events. Process analysis contributions have not only uncovered potentially harmful deviation between the designed process and the real process behavior, but also between the real process behavior and the event log (e.g. the notion of “invisible tasks”). In this section we propose the five criteria that will be considered in our event existence classification framework.

Firstly, two classification criteria will assist in determining the degree of *equivalence between the event log and the actual business process behavior*:

Actual business event: A real business event is an occurrence that happened in the organization’s business environment and that is deemed relevant for the organization.

Recorded event (in the event log): A recorded event is an occurrence that was registered by an information system of the organization in the event log under analysis, irrespective of the fact that it is a business event or not.

Process mining researchers and management scientists often assume the completeness of the event log (van der Aalst and Weijters, 2004). Concretely, this means that they assume that all possible behavior is present in the event log and that the execution of all event (and activities) will be precisely recorded. In real world projects these assumptions might be easily challenged, e.g. a manual task or a phone call will not necessarily be registered. Additionally, the event log may reflect a process behavior which deviates from reality, e.g. registration of non-executed activities or event records containing incorrect data due to anti-dating or employees using the authentication combination of other employees.

Secondly, while process-aware information systems may not be able to register every event relevant to the process under review, it might be possible to find evidence in *other sources*, such as event logs of related process, mail history, RFID registration systems, etc.:

Recorded event (in alternative source): A recorded event extracted from an alternative source is an occurrence that was registered by a system in a different data source, potentially but not exclusively another event log. Provided that you can link the events with the data in the alternative source.

Thirdly, the classification should take into account whether an event corresponds to the allowed or permitted mode of operation or is *deviating* from this mode. This type of criterion is strongly related to risk and compliance management.

Business rule compliance: An event is compliant with the business rules if, given the existing process history, the occurrence of the event does not violate any business rule.

Business rules are typically a representation of the business policies and the imposed external directives, e.g. regulation or industry standards. Compliance can be checked with logic based techniques, such as the linear temporal logic (LTL) checker (van der Aalst, De Beer and Van Dongen, 2005). Note that a business process can be a specific model of a set of business rules specifying specific dependencies between activities in the processes, e.g. activity precedence conditions. Therefore the business process can be considered as possible instances of the applicable business rule set.

Finally, we analyze whether an event is *supposed to happen* in the normal mode of operation of a business process.

Expected event: An event is expected when its presence is foreseen and anticipated during normal business operation, whereas its absence might result in a costly, deviating or special handling of the process instance.

For example the existence of a complete event related to an approval activity might be required and is therefore part of the normal mode of operation. An abortion event of a certain activity might be compliant with the business rule set but is probably not supposed to happen, and thus non-preferred or non-favored.

The next paragraph proposes an event existence classification framework based on these criteria and further elaborates on the different event types resulting from this classification.

PROPOSING THE EVENT EXISTENCE CLASSIFICATION FRAMEWORK

Based on the previously presented classification criteria we discerned thirteen event types (see Figure 1), some of them have not yet been recognized in the process analysis literature. As discussed earlier, the predominant focus of process analysis has been placed on analyzing the behavior present in the event log and under the assumption that the event log is a truthful representation of the real process behavior. Additionally, techniques for distinguishing wanted and unwanted behavior – for example conformance checking (Rozinat and van der Aalst, 2008) – are available. Consequently, these techniques deal with real-world business events that have been recorded. Research on artificially generated events on the other hand has focused on events for which no evidence of their existence can be found (Goedertier, Martens, Vanthienen and Baesens, 2009).

Actual Business Event	Recorded Event (in event log)	Recorded Event (alternative source)	Business Rule / Process Compliance	Event Category	Expected Event	Event Type	
Y	Y	-	Y	Truthful event	Y	Allowed preferred event	
			N		N	Allowed non-preferred event	
	N	N	N	N	Invisible event	Y	Impossible state
				-		N	Deviating event
				Y		Y	Recoverable event
				N		N	Invisible preferred event
N	Y	-	Y	False event	N	Invisible non-preferred event	
			N		Y	Impossible state	
	N	-	-	N	Unobserved event	N	Invisible deviating event
				Y		Y	False preferred event
				N		N	False non-preferred event
				Y		N	Impossible state
			Y		N	False deviating event	
			N		Y	Missing preferred event	
			Y		N	Unobserved non-preferred event	
			N		Y	Impossible state	
			N		N	Unobserved deviating event	

Figure 1: Event existence classification framework

The event existence classification framework broadens the spectrum of event types, by both proposing new and refining existing event types and criteria that might lead to new or improved applications of business process analytics. To further elaborate on the distinction and impact of each type, we first identify four *event categories* within the framework: truthful, invisible, false and unobserved events. These categories will be further elaborated in the following subsections.

Overview of the four Event Categories

Truthful Events

Truthful events are business events for which the business process analyst can find evidence of their existence, or – if sufficient – cannot find any counter evidence, and that were accurately recorded in the event log under review (Actual Business Event: Y and Recorded Event Y in Figure 1). Traditionally, this evidence collecting process is based on a reasonability principle, i.e. by establishing that the recording by the information system is done accurately and securely (e.g. match a sample of pay-complete events with bank account statements). Examples for the different event types belonging to this category can be found in Figure 2.

Event Type	Example
Allowed preferred event	Event for "Request for quotation" activity
Allowed non-preferred event	Event for "Cancel order" activity
Deviating event	Event for "Pay" activity before start event "Request for quotation"

Figure 2: Examples of truthful event types

Invisible Events

Invisible events are business events for which the business process analyst can find evidence of their existence but that were not recorded in the event log under consideration (Actual Business Event: Y and Recorded Event N in Figure 1). This typically involves manual tasks not performed in the context of the information system or activities performed under exceptional circumstances. *Recoverable events*: events that can be retrieved from alternative data sources, are considered as a special subset of invisible events. Examples of the invisible event types can be found in Figure 3.

Event Type	Example
Recoverable event	Event for "Warehouse move management" activity (RFID tracking system)
Invisible preferred event	Event for "Call supplier" activity (manual unregistered activity)
Invisible non-preferred event	Event for "Renegotiate order" activity (manual non registered activity, might result in an allowed non-preferred activity "Update order/orderline")
Invisible deviating event	Event for "Change invoice" activity (manual unregistered activity, mischarging e.g. change amount of cab-ride, number of people who went for lunch, etc.)

Figure 3: Examples of invisible event types

False Events

False events are business events for which the business process analyst can find evidence that counters the hypothesis that the event took place as described in the event log or, in strict environments, cannot prove that the event took place (Actual Business Event: N and Recorded Event Y in Figure 1). Nevertheless, these events have been recorded in the event log, which entails that every event with an event record in the log that is not a truthful representation of reality (e.g. antedated), must be considered as false. Although false events may be caused by bugs present in the information system which manages the process, they can also be the result of tampering or human-made errors, especially when the information system itself does not impose strict rules on the actual process flow. Examples of the different false event types can be found in Figure 4.

Event Type	Example
False preferred event	Event for "Issuance of tender" activity, registered activity but tender was never made publicly available
False non-preferred event	Event for "Cancel order" activity by originator not using his own security credentials
False deviating event	Event for "Pay" activity after a complete "Pay" activity had been done before (and the account number of last invoice was incorrect/not the number of the supplier)

Figure 4: Examples of false event types

Unobserved Events

Unobserved events are business events for which the business process analyst cannot find evidence of their existence and that are not recorded in the event log (Actual Business Event: N and Recorded Event N in Figure 1). Two main subcategories can be distinguished: unobserved events for which the occurrence is expected in normal mode of operation and therefore might imply a dangerous deviation and unobserved events for which an occurrence was not expected. The latter are mainly used for learning business rules from the behavior present in an event log. Examples of the different types of the unobserved event class can be found in Figure 5.

Event Type	Example
Missing preferred event	Event for "Approve order" activity that is not present in the log
Unobserved non-preferred event	Event for "Request credit note" activity that is not present in the log
Unobserved deviating event	Event for "Change order" activity after "Pay" Activity

Figure 5: Examples of unobserved event types

ILLUSTRATING CASE: DISCERNING EVENT TYPES IN THE SOFTWARE UPDATE AND DEBUG PROCESS OF A SOCIAL SECURITY SERVICE PROVIDER

For the purpose of illustrating the existence of the different event types in real business processes, we retrieved the event log recording behavior in an update (including new features) and debug process at a service provider in the social security sector. The event log was retrieved from the project management tool JIRA, which has been used for tracking the evolution in the update and debug cycles of an online reporting application. In total the event log contains 463 events scattered over 158 cases. The activity subset $A = \{\text{new feature business analysis, bug report analysis, information request, functional analysis, development, testing}\}$ and the originator set O encompasses 36 employees. Due to privacy concerns the originator information has been made anonymous. Figure 6 presents the Petri net obtained for the process event log with the $\alpha++$ miner in ProM (Wen, Wang and Sun, 2006).

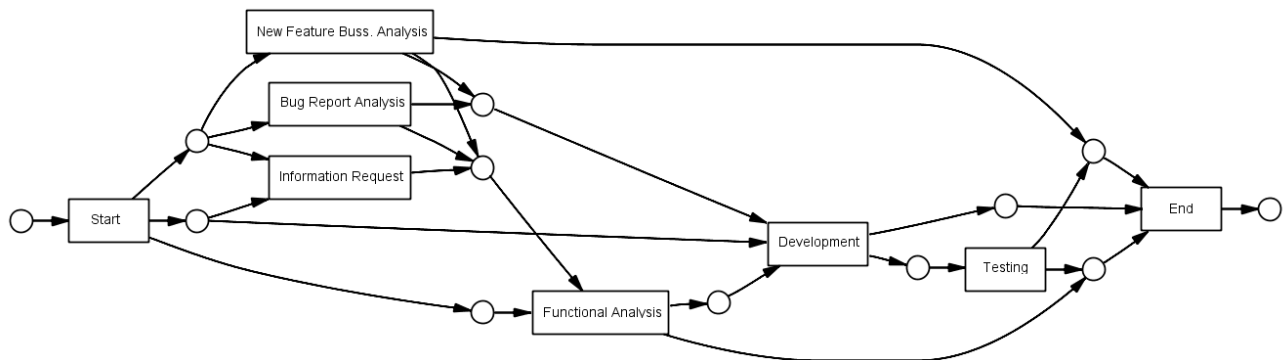


Figure 6: Excerpt of the mined process model for the software updating and debugging process of a social security provider

Table 1 provides an extract of the event log containing full process activity sequences for a limited number of instances. This event log extract in combination with additional information and/or other extracts will be used in the next section to demonstrate the existence of certain interesting event types.

Table 1: Extract of the event log for cases 17, 33, 59 and 99

Case ID	Activity	Originator	Timestamp	Type	Priority	Description
17	Start	Scheduler	15/11/2011	Bug	Critical	...
17	Bug Report Analysis	Laura	17/11/2011	Bug	Critical	...
17	Functional Analysis	Catherine	18/11/2011	Bug	Critical	...
33	Start	Scheduler	21/11/2011	New Feature	Critical	...
33	New Feature Buss. Analysis	Jeff	24/11/2011	New Feature	Critical	...
17	Development	Daniel	24/11/2011	Bug	Critical	...
17	Testing	Catherine	25/11/2011	Bug	Critical	...
17	End	Scheduler	25/11/2011	Bug	Critical	...
33	Functional Analysis	Elisabeth	30/11/2011	New Feature	Critical	...
33	Development	Richard	2/12/2011	New Feature	Critical	...
59	Start	Scheduler	2/12/2011	New Feature	Major	...
59	New Feature Buss. Analysis	Brian	6/12/2011	New Feature	Major	...
59	End	Scheduler	6/12/2011	New Feature	Major	...
33	Testing	Edward	7/12/2011	New Feature	Critical	...
33	End	Scheduler	7/12/2011	New Feature	Critical	...
99	Start	Scheduler	14/12/2011	Information Request	Critical	...
99	Information Request	Joseph	4/01/2012	Information Request	Critical	...
99	Functional Analysis	Catherine	6/01/2012	Information Request	Critical	...
99	Development	Mary	9/01/2012	Information Request	Critical	...
99	Testing	Joseph	10/01/2012	Information Request	Critical	...
99	End	Scheduler	10/01/2012	Information Request	Critical	...

Truthful Events

As well designed information systems and recording facilities register every event that happens in the business environment, we argue that the truthful events will count for the lion's share in an event log. Looking at the event types themselves, process instance 17 as described in the event log, for example, follows neatly the traditional software development process. Consequently, the instance complies with all business rules and as no counterevidence to their real-world existence was found, we classify them as allowed events.

Considering the "Expected"-criterion as well, the end event in instance 59 is an *allowed non preferred event type*, as it is allowed but unexpected and rather non-preferred. This process instance indicates that the service provider did not consider the implementation of the new feature, for example due to an unprofitable cost-benefit ratio. However, this course of action could negatively affect the providers image.

A typical example of a *deviating event type* (thus meaning Actual Business Event: Y and Recorded Event Y, but Rule/Process Compliant: N) in this event log is completion of the activity development, before at least one type of business analysis (and functional analysis) has been performed. These deviating event could be easily uncovered using querying tools such as the LTL-Checker in ProM (Clarke, Grumberg and Peled, 1999; van der Aalst, De Beer and Van Dongen, 2005). The following statement uncovers the existence of 26 instances (e.g. case 140) with this deviating event:

$$(\neg \text{development } W (\text{new_feature_business_analysis } \vee \text{bug_report_analysis } \vee \text{information_request}))$$

Meaning: development cannot occur until either a new feature analysis, bug report analysis or information request has been performed, without the restriction one of these three activities must occur at all.

Invisible Events

Capturing every single event important for a specific business process tends to be hard in a real-life setting. The update and debug process shown in Figure 1 is part of a larger issue management process, as depicted in Figure 7. The “Update Debug”-step thus conceals the process from Figure 1 as a sub process. Events related to the activities in this complete issue management process, e.g. Register Issue events, will thus not be recorded in the event log of the update and debug process. However, these events can then most likely be retrieved from the event logs of other process-aware information systems, so that these invisible events can be recovered (*recoverable event type*) and consequently handled in the same way as other truthful events.

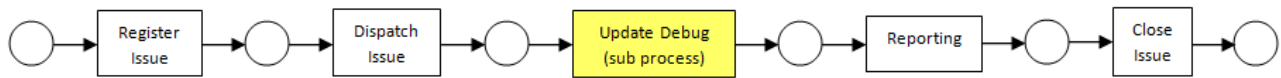


Figure 7: Complete issue management process

When an invisible event on the other hand cannot be retrieved from another data source, they remain hidden and cannot be analyzed further. However, their existence can sometimes be derived from the existence of correlated activities. The activity Dispatch Issue in Figure 7, for example, is a manual task and therefore not registered in an event log. However, when it can be observed that the Update Debug sub process has taken place, it is reasonable to assume that the dispatching of the issue itself took place as well.

False Events

False events appear to be legitimate to the process analyst, but do not provide a truthful reflection of real process behavior. For example, the testing event of case 99 appears compliant with the business rules (i.e. after a software development activity has been executed a testing activity must be performed) taking into account the process instance history. However, the following record of a human resource system keeping track of absence amongst employees (see Table 2), provides counterevidence¹. As the tester was on sick leave it would have been impossible for him to perform the testing activity. Consequently, the testing complete event of case 99 is a *false preferred event*.

Table 2: Record extract from a HR system keeping track of employee absence

ID	Employee	Type of Leave	From	To
52	Joseph	Sick leave	09/01/2012	13/01/2012

Unobserved Events

Well-designed software development processes always require the execution of testing activities after update development or debugging activities have been performed. Consequently, these testing events are expected in the normal mode of operation. Using the ProM LTL-Checker in combination with the LTL statement below, we were able to find 109 *missing preferred testing events*.

$$\square(\text{development} \rightarrow \diamond \text{testing})$$

While missing preferred events can indicate important issues, investigating other unobserved events may result in valuable information on process dynamics. In order to take these unobserved events into account during the process analysis, artificial negative events are generated and injected in the event log, using an induction technique described in (Goedertier, Martens, Vanthienen and Baesens, 2009). Table 3 presents the partial event trace for case 33 with injected artificial negative events, so that it is observed that a testing event was unobserved in the context of this process instance, before a functional analysis. This event thus represents an *unobserved deviating event*, which corresponds, of course, with business desires, as non-

¹ The record from the HR system is fictional and purely for illustrative purposes.

allowed, deviating events should indeed preferably remain unobserved. Further details can be found in the section on artificial event generation research.

Table 3: Partial event trace for case 33 supplemented with artificial negative events

Case ID	Activity	Originator	Generated Event	Timestamp	Type	Priority	Description
33	New Feature Buss. Analysis	Jeff		24/11/2011	New Feature	Critical	...
33	<i>Start</i>		<i>artificial negative</i>		<i>New Feature</i>	<i>Critical</i>	...
33	<i>Information Request</i>		<i>artificial negative</i>		<i>New Feature</i>	<i>Critical</i>	...
33	<i>Bug Report Analysis</i>		<i>artificial negative</i>		<i>New Feature</i>	<i>Critical</i>	...
33	<i>New Feature Business Analysis</i>		<i>artificial negative</i>		<i>New Feature</i>	<i>Critical</i>	...
33	<i>Testing</i>		<i>artificial negative</i>		<i>New Feature</i>	<i>Critical</i>	...
33	Functional Analysis	Elisabeth		30/11/2011	New Feature	Critical	...

OBSERVATIONS FOR BUSINESS PROCESS ANALYTICS RESEARCH AREAS

The event existence framework enables a better orientation of the different business process analytics research areas and their applicability in terms of event types. Additionally, this section aims to indicate potential issues in the research fields.

Process Mining Research

Process mining as a research area can be situated at the crossing of data mining and business process management. Whereas the field shares its goal with data mining (i.e. learning from large data repositories) (Tan, Steinbach and Kumar, 2005), it can be best situated in the diagnosis cycle of business process management (vom Brocke and Rosemann, 2010). Process mining techniques are generally divided into three categories: process discovery, compliance & conformance checking and process enhancement & extension (van der Aalst, ter Hofstede, Weijters, and Weske, 2003). The first category focuses on the induction of formal models from given event logs, without assuming the presence of an a-priori model (Cook and Wolf, 1998; de Medeiros, van Dongen, van der Aalst and Weijters, 2004; Goedertier, Martens, Vanthienen and Baesens, 2009; van der Aalst, de Beer and van Dongen, 2005; van der Aalst, Weijters and Maruster, 2004; Weijters, van der Aalst and de Medeiros, 2006). Secondly, compliance and conformance checking deals with analyzing process behavior that deviates from a prescriptive model or a set of business rules (Rozinat and van der Aalst, 2008; van der Aalst, de Beer and van Dongen, 2005). The final category tries to further enhance current prescriptive process models, based on properties and behavior found in actual recorded data.

As event logs are the primary information source for process mining based analysis, it is immediately clear that the applicability of traditional process mining techniques is limited to the events recorded in event logs, regardless of the fact that they correspond with real business events or not (see Figure 1). When a recorded, logged event does match with a business event, process discovery, conformance checking and enhancement techniques can all be applied without any direct danger of reaching skewed or incorrect conclusions. Well-known conformance and rule checking methods can then safely be applied to distinguish between allowed and deviating events. We refer to (Rozinat and van der Aalst, 2008; van der Aalst, de Beer and van Dongen, 2005) for a detailed overview of such techniques.

On the other hand, as seen in the illustration above, when a log entry does not match with a real business event, applying conformance checking or other process mining tasks will lead to wrong findings, as the event log contains *false events* which did not occur in real life.

Artificial Event Generation Research

Recently, scholars have shown that process mining tasks are currently limited to the harder setting of supervised learning as information about events that were prevented or prohibited from taking place is often unavailable in real-life event logs and

consequently cannot guide mining tasks (Ferreira and Ferreira, 2006; Goedertier, De Weerd, Martens, Vanthienen and Baesens, 2011; Goedertier, Martens, Baesens, Haesen and Vanthienen 2008; Goedertier, Martens, Vanthienen and Baesens, 2009). Such negative information can, however, be useful to discover the discriminating properties of the underlying process. In some cases, however, process logs do naturally contain negative events. Access logs, for example, contain information about users that have obtained authorization and information about workers who were refused authorization to perform a particular task. In many other cases, information systems do not reveal such information in terms of negative events. For instance, when a Workflow Management system creates a number of tasks and assigns them to several users, it will not explicitly expose the tasks it did not create or provide information about users to which it could not assign the item.

Nevertheless, we can still apply an induction algorithm developed by Goedertier et al. (Goedertier, Martens, Vanthienen and Baesens, 2009) in order to supplement event logs with artificially generated negative events, by testing at each event in a given trace if a certain other event of interest could also occur at the position being looked at. Applying this technique in our context, artificially inducing events based on process history provides a fitting tool in order to detect which events were never observed at certain times in certain process instances. By examining these “under-represented” events and matching them with the set of current business rules and designed process model, investigators and analysts can then spot non-occurred events which were, however, allowed and perhaps even desired. Such cases provide a good indication to investigate further in order to uncover the root causes behind this missing behavior, especially if the absence of a certain set or sequence of events is overshadowed by the presence of another set (meaning that, for some reason, certain perfectly valid alternatives are more or less ignored during day-to-day business conduct in favor of other activity paths).

Applying the induction technique as described above thus allows us to uncover and analyze another group of events in our classification framework, namely those which did not occur in real-life (representing rare or infrequent tasks): the *unobserved events*.

Event Log Merging Research

An important quadrant in the event existence classification consists of the non-recorded invisible events. Whereas this often deals with manual or exceptional activities, it has been argued that an important share of the business processes are supported by multiple information systems making it difficult to trace the process history (Georgakopoulos and Hornick, 1995). Moreover, the same contribution indicated that there is often no clear link between the process parts of specific process instances. We argue however that it is possible, although admittedly difficult in some cases, to recover a great deal of information from other sources; a wealth of relevant process information can nowadays be retrieved from alternative data sources (e.g. RFID tracking systems).

Event log merging research is confronted with a double-sided problem: finding both event-process instance matches and event log merging. Whereas the former focuses on determining the process instance to which the individual events belong, the latter deals with reconstructing the exact activity trace. Discovering the corresponding process instance can be translated to a data matching problem. (Claes and Poels, 2011; Salinesi, Pastor, Claes and Poels, 2011) propose respectively an artificial immune system and a genetic algorithm technique in order to do so. Other event correlation techniques have been presented in (De Pauw, Hoch and Huang, 2007; Motahari-Nezhad, Saint-Paul, Casati and Benatallah, 2010). The trace reconstruction task has to deal with timestamp reliability and differences in timestamp precision.

Process-Oriented Auditing, Compliance and Conformance Checking Research

From an auditing and compliance perspective business process analytic techniques and tools present a potentially important opportunity (Green, Best, Indulska and Rowlands, 2005). After all, these tools and techniques provide the auditor or compliance officer with a unique insight of the real behavior in the organizations behavior. Three main categories of business process analytics/mining techniques useful in this context have been uncovered: process discovery (Cook and Wolf, 1998; de Medeiros et al., 2004; Goedertier et al., 2009; van der Aalst et al., 2005; van der Aalst et al., 2004; Weijters et al., 2006), conformance checking & delta analysis (Rozinat and van der Aalst, 2008) and property verification (van der Aalst, de Beer and van Dongen, 2005).

While these techniques can provide high levels of assurance in internal/process control effectiveness as well as provide strong evidence for control ineffectiveness, precaution should be taken. We identified two major problems in this context: issues with traceability and auditability and the existence of fraudulent events. The former is a consequence of the inability to capture all the events relevant for a specific process instance. Consequently, certain business rules might seem to be violated when tested on the event data. The latter deals with seemingly legitimate events, but in fact they did not occur/not happened as described in the event log. Such entries might indeed potentially conceal fraudulent behavior, both in the case when the false event is allowed by the business model and constraints in place (this corresponds to a fraudulent execution), but also –

and not less so – when the false recorded event is not allowed by business policies and rules. Although such events might be result of “innocent” system failures or implementation faults (worth looking into) it might also be the case that these suspicious events must be interpreted as forgery.

Event Type	Audit Relevance
Truthful allowed event	No negative evidence for internal/process control effectiveness
Truthful deviating event	Evidence for a lack of internal/process control effectiveness
Invisible event	Issues with traceability and auditability
False event	Fraudulent events
Missing preferred event	Evidence for a lack of internal/process control effectiveness
Missing non-preferred or deviating event	No negative evidence for internal/process control effectiveness

Figure 8: Audit relevance of each event type

CONCLUSION

This paper presented an event existence classification framework based on the following business criteria: business event, recorded event in event log/ alternative source, business rule compliance and expected event. This resulted in the identification of thirteen interesting event types distributed over four categories: namely truthful, invisible, false and unobserved events. Additionally, we situated the applicability of the different business process analytics research areas and indicated the potential issues.

REFERENCES

1. Buhl, H., Ruglinger, M., Stoeckl, S. and Braunwarth, K. (2011) Value Orientation in Process Management Research Gap and Contribution to Economically Well-Founded Decisions in Process Management, *Business & Information Systems Engineering*, 3, 3, 163-172.
2. Claes, J. and Poels, G. (2011) Merging computer log files for process mining : an artificial immune system technique, in *9th International Conference on Business Process Management (BPM - 2011)*, Clermont-Ferrand, France, 12.
3. Clarke, E., Grumberg, O. and Peled, D. (1999) Model Checking, The MIT Oress, Cambridge, Massachusetts and London.
4. Cook, J. and Wolf, A. (1998) Discovering models of software processes from event-based data, *ACM Transactions on Software Engineering and Methodology*, 7, 3, 215-249.
5. de Medeiros, A., van Dongen, B. and van der Aalst, W. (2004) Process mining: Extending the alpha-algorithm to Mine Short Loops, *Technical report, TU Eindhoven*.
6. De Pauw, W., Hoch, R. and Huang, Y. (2007) Discovering Conversations in Web Services Using Semantic Correlation Analysis, in *IEEE International Conference on Web Services (ICWS 2007)*, July, 639-646.
7. Ferreira, H. and Ferreira, D. (2006) An integrated life cycle for workflow management based on learning and planning, *International Journal of Cooperative Information Systems*, 15, 4, 485-505.
8. Georgakopoulos, D., Hornick, M. and Sheth, A. (1995) An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, *Distributed and Parallel Databases*, 3, 2, 119-153.
9. Goedertier, S., De Weerd, J., Martens, D., Vanthienen, J. and Baesens, B. (2011) Process discovery in event logs: An application in the telecom industry, *Applied Soft Computing*, 11, 2, 1697-1710.
10. Goedertier, S., Martens, D., Baesens, B., Haesen, R. and Vanthienen, J. (2008) Process mining as first-order classification learning on logs with negative events, in *Business Process Management Workshops*, 42-53.
11. Goedertier, S., Martens, D., Vanthienen, J. and Baesens, B. (2009) Robust Process Discovery with Artificial Negative Events, *Journal of Machine Learning Research*, 10, 1305-1340.

12. Green, P. F., Best, P., Indulska, M. K. and Rowlands, T. P. (2011) Information systems audit and control issues for enterprise management systems: Qualitative evidence, *Australian Accounting Review*, 15, 3, 67-77.
13. Motahari-Nezhad, H. R., Saint-Paul, R., Casati, F. and Benatallah, B. (2010) Event correlation for process discovery from web service interaction logs, *The International Journal on Very Large Data Bases*, 20, 3, 417-444.
14. Rozinat, A. and van der Aalst, W. (2008) Conformance checking of processes based on monitoring real behavior, *Information Systems*, 33, 1, 64-95.
15. Salinesi, C., Pastor, O., Claes, J. and Poels, G. (2011) Integrating computer log files for process mining : a genetic algorithm inspired technique, in Camille Salinesi and Oscar Pastor (Eds.) *International conference on Advanced Information Systems Engineering (CAiSE '11)*, London, UK.
16. Tan, P.-N., Steinbach, M. and Kumar, V. (2005) *Introduction to Data Mining*, Addison-Wesley.
17. van der Aalst, W. and Weijters, A. (2004) Process mining: a research agenda, *Computers in Industry*, 53, 3, 231-244.
18. van der Aalst, W., Weijters A. and Maruster, L. (2004) Workflow mining: discovering process models from event logs, *IEEE Transactions on Knowledge and Data Engineering*, 16, 9, 1128-1142.
19. van der Aalst, W., De Beer, H. and Van Dongen, B. (2005) Process mining and verification of properties: An approach based on temporal logic, *Information Systems Journal*, 3760, 130-147.
20. van der Aalst, W., Reijers, H., Weijters, A., van Dongen, B., Alves de Medeiros, A., Song, M. and Verbeek, H. (2007) Business process mining: An industrial application, *Information Systems*, 32, 5, 713-732.
21. van der Aalst, W., ter Hofstede, A., Weijters, A. and Weske, M. (2003) Business process management: A survey, *Business Process Management*, 1-12.
22. vom Brocke, J. and Rosemann, M. (2010) *Handbook on Business Process Management: Strategic Alignment, Governance, People and Culture*, Springer, Berlin, Germany.
23. Weijters, A., van der Aalst, W. and de Medeiros, A. (2006) Process mining with the heuristics miner-algorithm, *Technical report, TU Eindhoven*.
24. Wen, L., Wang, J. and Sun, J. (2006) Detecting Implicit Dependencies Between Tasks from Event Logs, *Frontiers of WWW Research and Development*, 591-603.
25. zur Mühlen, M. and Shapiro, R. (2010) Business process analytics, in *Handbook on Business Process Management: Strategic Alignment, Governance, People and Culture*.