Enriching Raw Events to Enable Process Intelligence – Research Challenges

Nico Herzberg, Mathias Weske

Technische Berichte Nr. 73

des Hasso-Plattner-Instituts für
Softwaresystemtechnik
an der Universität Potsdam

Technische Berichte des Hasso-Plattner-Instituts für
Softwaresystemtechnik an der Universität Potsdam

Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam | 73

Nico Herzberg | Mathias Weske

Enriching Raw Events to Enable Process Intelligence

Research Challenges

Universitätsverlag Potsdam

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.de/ abrufbar.

Universitätsverlag Potsdam 2013
http://verlag.ub.uni-potsdam.de/

Am Neuen Palais 10, 14469 Potsdam
Tel.: +49 (0)331 977 2533 / Fax: 2292
E-Mail: verlag@uni-potsdam.de

Die Schriftenreihe **Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam** wird herausgegeben von den Professoren des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam.

ISSN (print) 1613-5652
ISSN (online) 2191-1665

Online veröffentlicht auf dem Publikationsserver der Universität Potsdam
URL http://pub.ub.uni-potsdam.de/volltexte/2013/6401/
URN urn:nbn:de:kobv:517-opus-64012
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus-64012

Zugleich gedruckt erschienen im Universitätsverlag Potsdam:
ISBN 978-3-86956-241-4

Enriching Raw Events to Enable Process Intelligence – Research Challenges

Nico Herzberg and Mathias Weske

Hasso Plattner Institute at the University of Potsdam
Prof.-Dr.-Helmert-Strasse 2-3, 14482 Potsdam
{nico.herzberg,mathias.weske}@hpi.uni-potsdam.de

Abstract. Business processes are performed within a company's daily business. Thereby, valuable data about the process execution is produced. The quantity and quality of this data is very dependent on the process execution environment that reaches from predominantly manual to full-automated. Process improvement is one essential cornerstone of business process management to ensure companies' competitiveness and relies on information about the process execution. Especially in manual process environments data directly related to the process execution is rather sparse and incomplete. In this paper, we present an approach that supports the usage and enrichment of process execution data with context data – data that exists orthogonally to business process data – and knowledge from the corresponding process models to provide a high-quality event base for process intelligence subsuming, among others, process monitoring, process analysis, and process mining. Further, we discuss open issues and challenges that are subject to our future work.

1 Introduction

Nowadays many companies face a competitive market environment and therefore are managed in a process-oriented fashion. Business process management (BPM) combine concepts, methods and techniques to support the design, administration, configuration, enactment, and analysis of business processes [23]. Central to business process management are process models, as they explicitly describe the operations that need to be carried out and are used, among others, for documentation, certification, and enactment. For improving these business process models it is essential to monitor and analyze the execution of these processes to identify the weak points in the process. Prof. Alejandro Buchmann stated in an interview, conducted at 10th International Conference on Business Process Management 2012, that "rearchitecting of business process management systems to handle monitoring aspects and complex event processing better and to do that in a very natural way" is one of two very interesting areas researchers should focus on in the next years.

During the execution of business processes – operating process instances – several events, in concrete real-world happenings, occur that are valuable for gaining insights about the processes and their execution. This information about events is essential for business process intelligence applications such as process monitoring (the monitoring of running process instances) and analysis (the analysis of completed process instances) [16,7]. Business process intelligence is the combination of techniques and methods from business intelligence and business process management aiming at analysis and improvement of processes and their management. Business intelligence summarizes all technologies and methods to well-arranged information to support decision making [22]. Depending on the degree of automation of process execution the number of the observation and representation of these events reaches from rather sparse in non-automated process execution environments, e.g., in health care treatment processes, over more frequently in semi-automated process execution environments, to complete available for every process instance, e.g., for processes executed by a process engine.

Besides the number of observed and represented events the quality of the representation is varying from rather poor, e.g., events are recorded very abstract on paper, to excellent, e.g., events are recorded in a well-defined form and their context information including process relation is represented completely. The Process Mining Manifesto [22] defines five event log maturity levels that we break down for single events. The levels reach from one star which stands for poor quality to five stars that indicates high quality event representation. Setting up a meaningful process intelligence application requires a fair number of events, that provide high information content. Thus, the goal is to utilize as much information about events as possible.

In this paper we focus on the challenge of improving event content and present an holistic approach, combining methods and techniques. The information content of single event representations is improved by (a) utilizing the knowledge provided in process models and (b) embedding context data that exist orthogonal to business process data. This is challenging since information about events and context data usually do not contain obvious business process correlation information. Thus, pointing existing information to the right process execution is not trivial. Correlating event representations to the respective process activities range from simple, e.g., label matching of events and activities, to very complex, e.g., an event triggered in a process provides progress information of another process. We list the challenges that the approach implicate to build up an agenda for our further research.

The remainder of the paper is structured as follows. In Section 2, we introduce a motivating example of a process-oriented cinema and derive the problems in the area of event quality and correlation. Afterwards the approach is sketched as an overview, based on which the necessary techniques are introduced and the methodology of the approach is explained in detail, in Section 3. In Section 4, the approach is applied to the motivating example and the challenges that need to be solved are listed and explained in detail in Section 5. In Section 6, a selection of related work is discussed, before we conclude the paper and give an outlook on how the topic is developed further in Section 7.

2 Motivating Example and Problem Statement

Organizations following BPM have a clear understanding of their major processes and describe those in an adequate form. These descriptions build the blueprint for the business process execution during which several events happen. Information about the major business processes are expressed as BPMN diagrams for documentation reasons, for instance. In this paper, we refer to an example of a process-driven cinema called SuperMovies. Each movie show follows the designed process model shown in Figure 1. The process “Movie Show” describes the activities to be performed before, during and after a movie show including feedback gathering. After several commercials are shown the movie will be presented. Once the movie ended the cleaning staff will wait for 15 minutes, so that the visitors can leave, before they start cleaning up the cinema hall. SuperMovies initiated a quality campaign to gather visitor‘s feedback regarding the shown movies. This activity is reflected in the documentation of the business process in parallel to the cleaning activity.

For gathering visitor‘s feedback the cinema operator installed only one voting system for the whole cinema. To gain a lot of feedback the voting system is kept very simple and consists only of three buttons of color green, indicating the movie was very good, yellow, the movie was okay, and red, the movie was bad. The information created by this voting system is stored in a database. For each button push a data record is written to the database including a timestamp and the color of the pushed button, see Table 1. The voting is an event and it is created while the business process “Movie Show” is executed.

An *event* happens in a particular *point in time* at a certain *place* in a certain *context*. The context describes the situation in which the event happened. Some of these events are represented in information systems to which we refer as event objects. An event object is an object that represents, encodes, or records an event, generally for the purpose of computer processing [12]. These event objects provide information about the business process execution. The Event Processing Technical (EPTS) states that “the word *event* is overloaded so that it can be used as a synonym for event object. In discussing event processing, the word *event* is used to denote both the every day meaning (anything that happens) and the computer science meaning (an event object or message)” [12]. The event

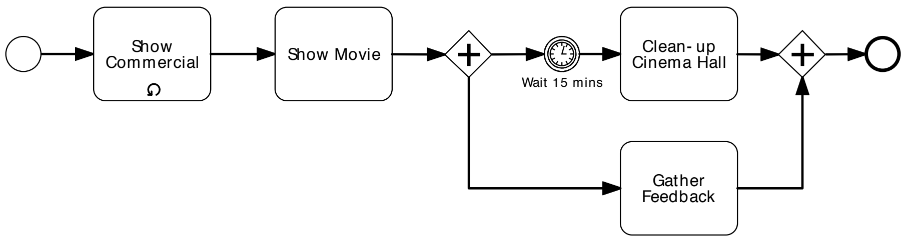

Figure 1. Process “Movie Show” describing the activities to be performed during a movie show including feedback gathering

ID	Timestamp	Rating
...	...	...
101	2012-09-10 22:10:03	green
102	2012-09-10 22:13:08	yellow
103	2012-09-10 22:14:13	green
104	2012-09-10 22:16:55	green
105	2012-09-10 22:17:01	red
106	2012-09-10 22:17:42	green
107	2012-09-10 22:17:59	green
108	2012-09-10 22:19:00	yellow
109	2012-09-10 22:19:02	yellow
110	2012-09-10 22:19:24	red
111	2012-09-10 22:21:54	green
112	2012-09-10 22:22:14	green
...	...	...

Table 1. Excerpt of the database logging the signals from the voting system (rating.db)

Hall	Timestamp	Status
...	...	...
3	2012-09-10 19:15:00	100
3	2012-09-10 19:30:00	50
3	2012-09-10 19:39:00	0
2	2012-09-10 19:42:54	100
1	2012-09-10 19:58:02	100
2	2012-09-10 20:00:05	50
2	2012-09-10 20:06:32	0
1	2012-09-10 20:15:00	40
1	2012-09-10 20:22:05	0
3	2012-09-10 21:35:00	100
2	2012-09-10 21:44:42	100
1	2012-09-10 22:09:23	100
...	...	...

Table 2. Excerpt of the database logging the lighting status (lighting.db)

objects introduced in this paper are explicitly named as such and we stick to event as the real-world happening.

For example, the cinema visitor Bob pushes the voting button to rate the movie "Rambo X" after he saw it. The point in time is for example "September 10th, 2012 22:14:13" and the place the event happened is "in front of the cinema halls". The event context could be described as follows: "Visitor Bob saw Rambo X in cinema hall 1 and liked it very much", so he pushed the green button. This event is represented in the information system as an event object.

While representing the happened event as an event object in an information system, in most cases information about the event itself gets lost. This is due to the fact that the event is represented in an abstract form that does not contain all information about an event available.

The representation of the events of voting a movie is subject to *information loss* as well. The event object representing the voting of Bob does not contain any information about the movie rated, the visitor's name, or the cinema hall the movie was shown, see Table 1 – ID 103. The information represented in the event object is the time when a rating was done and the rating itself.

Information content resp. completeness is, among others, one criteria for event quality. For classifying the quality of event objects, we refer to the five maturity levels presented in the Process Mining Manifesto [22] and adopt them to the single event object level, see Table 3. The levels reach from one star, which stands for poor quality, to five stars that indicates high quality event objects.

Considering our example of event objects of the voting system, these can be classified as 3-star event objects, because the reality match is guaranteed by the system itself, i.e., once someone pushes a button, exactly one event is recorded by the voting system. But the reference to the process instance is not given. Most often in manual business process executing environments, 3-star event objects

Level	Characterization	Example
*****	Highest level: the event object is of excellent quality (i.e., trustworthy and complete) and well-defined. The event objects are recorded in an automatic, systematic, reliable, and safe manner. Privacy and security considerations are addressed adequately. Moreover, the event objects recorded and all of their attributes have clear semantics. This implies the existence of one or more ontologies. Event objects and their attributes point to this ontology.	Semantically annotated event objects of BPM systems.
****	Event objects are recorded automatically and in a systematic and reliable manner, i.e., event objects are trustworthy and complete. Unlike the systems operating at level ***, notions such as process instance (case) and activity are supported in an explicit manner.	Event objects of traditional BPM workflow systems.
***	Event objects are recorded automatically, but no systematic approach is followed to represent events. However, unlike event objects at level **, there is some level of guarantee that the event objects match reality (i.e., the event objects are trustworthy but not necessarily complete). Consider, for example, the event objects recorded by an ERP system. Although events need to be extracted from a variety of tables, the information can be assumed to be correct (e.g., it is safe to assume that a payment recorded by the ERP actually exists and vice versa).	Event objects in ERP systems, event objects of CRM systems, event objects of messaging systems, event objects of high-tech systems.
**	Event objects are recorded automatically, i.e., as a by-product of some information system. Coverage varies, i.e., no systematic approach is followed to decide which event objects are recorded. Moreover, it is possible to bypass the information system. Hence, event objects may be missing or not recorded properly.	Event objects of document and product management systems, event objects of embedded systems, worksheets of service engineers.
*	Lowest level: event objects are of poor quality. Event objects may not correspond to reality and may be missing. Event objects that are recorded by hand typically have such characteristics.	Events represented in paper documents routed through the organization ("yellow notes"), paper-based medical records.

Table 3. Event object quality levels, cf. [22]

are the highest quality available, because the correlation to the concrete process execution is not supported explicitly. Nonetheless, the aim must be to bring them at least to the fourth level by enriching them with further information, to approach completeness, and more important with information about the related business process.

Besides the quality campaign, the cinema SuperMovies is running a sustainability initiative to reduce the electricity consumption. Therefore, in every cinema hall the lighting is monitored to improve the energy consumption. Every change in the lighting status is recorded in a database, see Table 2. The lighting status is stored as a number in percent to capture dimming phases also. 0 % means the light is off, 100 % means light is on full power, a value in between means that the light is dimmed to a certain degree.

Further, organizations are in possession of data that are not related to a process execution but valuable as well to gain insights about a organizations performance. We refer to such kind of data as context data. Context data exists orthogonal to a business process. It contains information about the correlation of events occurring during process execution and the corresponding business process, and provides the business context.

The cinema SuperMovies is planning their movie shows on a weekly basis for all of their three cinema halls and store the program in a spread sheet on a file server, see Table 4. This program data exist independent from business process related data and we refer to it as context data.

Looking at the data of the cinema one can see that they are not connected in any way, see Figure 2. The data exist in silos and only deliver value for the

Movie	Start	Duration	Cinema Hall
Rambo X	2012-09-10 20:15:00	107	1
Rambo X	2012-09-10 22:30:00	107	2
Rambo X	2012-09-11 20:00:00	107	1
Rambo X	2012-09-12 20:15:00	107	1
Rambo X	2012-09-12 22:30:00	107	2
...	...	...	...
Bad Boys 3	2012-09-10 20:45:00	98	2
Bad Boys 3	2012-09-10 22:15:00	98	3
Bad Boys 3	2012-09-11 20:00:00	98	2
Bad Boys 3	2012-09-12 20:00:00	98	2
...	...	...	...
Ice Age 5	2012-09-10 19:30:00	116	3
Ice Age 5	2012-09-11 19:30:00	116	3
Ice Age 5	2012-09-12 19:30:00	116	3
...	...	...	...

Table 4. Excerpt of the cinema program for calender week 37/2012 (programCW37.xls)

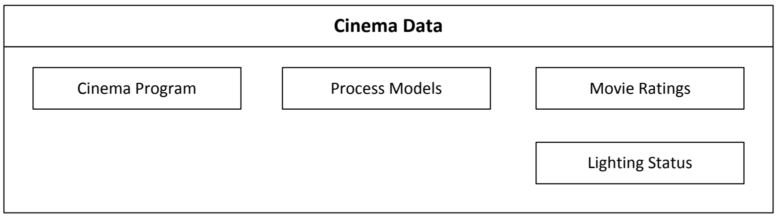

Figure 2. Data in a cinema with originally no interconnections

applications of the specific systems, namely the process modeller, cinema program planning system, voting system, and the lighting monitoring system.

This leads us to the next issue of missing *process context*. Part of process intelligence is process monitoring and process analysis, which are performed based on concrete process executions, so-called process instances. We refer to process monitoring as the monitoring of running process instances, whereas process analysis is the analysis of completed process instances (separate or summarized). This requires that the available information about an event, context data, and process knowledge is interconnected to these business process instances. The event objects in most cases do not contain any information about the relation to one or several business processes (they are 3-star event objects maximum). The event objects are recorded because of a certain event happened, not knowing that the event occurs because a business process is executed.

Additionally, it could be valuable to gain further information from utilizing event objects that were originally intended for another process. The events about the lighting, originally intended by the sustainability processes, could give information about the progress of the process "Movie Show". When showing the commercials the lighting is dimmed only, but during the movie itself switched off completely. Thus, the start of the movie is represented by the event object indicating that the light is switched off in the cinema hall.

3 Event Enrichment Framework

As stated in Section 2, organizations following BPM are in possession of information about their processes, the process models, data about the execution of these processes, so-called event objects, and data that is not directly correlated to a business process, so-called context data. The event objects within the companies' information system landscape are very valuable for gaining insights about the processes and their execution. We refer to event objects that capture events in a raw format as raw events.

Definition 1 (Raw Event).
A raw event $E_R = (raw\_data)$ is a representation of an event in a data source, where raw_data is the available data of the event. raw_data does not have a specific form. ◇

A data source could be a sheet of paper, a real-world object or an information system. For this paper we restrict to information systems, as the medium holding information about a raw event. The EPTS is referring to a representation of a real-world happening as a raw event as well [12].

Thus, they should be used as basis for process intelligence. As sketched in Section 2, raw events are at most times subject to information loss and are not related to a process context. To overcome these issues, we present in this section an overview of the approach that combines the raw events with knowledge from the available process models and context data to enrich the event content, bring them into business resp. process context, and associate them with each other.

The goal of the raw event processing is providing high-quality events, so-called process events for process event consumers, such as process monitoring, process analysis, process mining tools. Process events contain information that originate from raw events enriched by context data and correlated with information about the process they belong to. Process events are event objects of highest possible quality that could be reached by the presented approach. Figure 3 shows the big picture how process models, raw events, and context data come together to form process events by applying certain methods and techniques.

The approach for forming process events is based on process models. The definition of a process model used in this work subsumes a number of nodes N and their execution order F, as shown in Definition 2. This covers commonly used process modeling languages, such as BPMN [17] and EPC [10], but also value

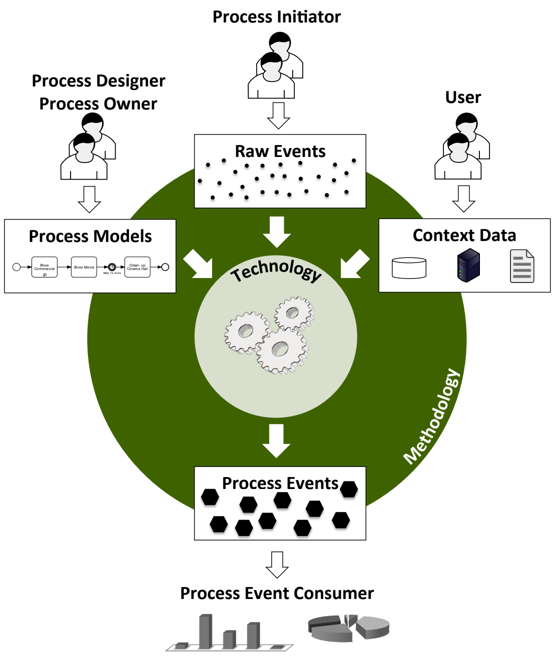

Figure 3. Concept Overview

chains [18], where N represents coarse grained units of work and F represents the execution order of these work units. In BPMN for instance, N is partitioned into *activities*, *gateways*, and *events*, whereas F represents sequence flow among these nodes.

Definition 2 (Process Model).
A process model is a tuple $P = (N, F)$, where N is the set of nodes and $F \subseteq N \times N$ is the control flow relation that captures ordering constraints of the nodes for process execution. $\diamond$

With respect to our introduced example, cf. Section 2, this definition holds for the process model for movie shows, cf. Figure 1.

To enrich the event content on several event processing levels, context data could be utilized, to overcome the issue of information loss, see Section 2. In the example of the cinema SuperMovies, the weekly cinema program could be seen as context data for utilizing the rating results for process intelligence for instance. The cinema program exists in parallel to the process "Movie Show" and is not interconnected to it, nevertheless it contains information that supports event correlation. However, the cinema program could be the output of another business process, i.e., the cinema program planning process. Thus, the cinema program is not context data for the process of cinema program planning. Available context data needs to be correlated with the respective business processes also. The knowledge about the process came with the process models the individual process instances follow.

With the presented approach we address all kinds of process execution environments, namely complete manual execution, semi-automated execution, as well as automated execution. We do not restrict to one of them, however, because of the absence of a central event controlling and logging instance, e.g., a process engine, the described techniques and methods will support manual process execution environments most.

In the remainder of this section, we will introduce basic operations for event processing, in Section 3.1. In Section 3.2, the methodology of the framework is described and the single stages of the framework are explained. Afterwards, a brief overview about process event consumption is given, in Section 3.3, before we summarize the framework in Section 3.4.

3.1 Technology

Constructing process events from raw events requires several event processing operations.

Aggregation is the bundling of event objects of the same event object type to one resulting event object [5]. For example, all ratings to a movie are combined to one event indicating whether the movie was rather good or bad.

Composition is the bundling of event objects of different event types to one resulting event object [5]. The event object of switching on the light and identifying the first voting event object can be composed to an event object stating the end of the movie.

Correlation is the functionality to relate different event objects to a joint instance. For instance, the rating from different movie shows is correlated to a certain movie to get an indication whether the public likes or dislikes it.

Enrichment is the translation of an event object into another by adding information to them [5]. For example, adding the movie name to an event representing a rating would be an enrichment of the original raw event.

Projection is removing information from the event content during translation of an event object into another [5]. In the case, the number of the ratings is of interest only, the rating values, namely green, yellow, or red can be cut of.

3.2 Methodology

The framework describes several event object processing phases. (i) First of all, occurred events need to be represented in the companies' information systems. This is not in scope of this paper, but we describe it briefly for completeness reasons. (ii) In the next step, the raw events need to be extracted from these information systems, brought into a common form (normalization) and stored in an event store. (iii) Afterwards, the business context is added by combining normalized events with context data. (iv) In the last step, the knowledge about the processes is incorporated and process events are formed. Figure 4 shows the phases (i-iv) mentioned above, their outcome, their interdependency, and the relations of all event objects. In the latter three phases (ii-iv) different structured events based on structured event types are used. A structured event E is a derivation of an event object, holding information such as an identifier *id*, a *timestamp*, and the *event content* C. The event content is structured and could be a set of key-value-pairs for instance, but also a tree-structure, e.g., expressed in extensible markup language (XML). A structured event object type ET describes a class of structured events that have the same form and origin. Besides structured event specific attributes, a structured event type has a *content description* that holds the structure of the event content of a structured event, e.g., by defining the attributes (keys) or by an XML Schema Definition (XSD). The *meta-data* specify the structured event type to allow a better handling of the structured event types, e.g., for finding those in a structured event type repository.

In the remainder of this section, we will describe the representation, normalization, business event creation, process event creation phase in detail.

3.2.1 Representation

At the beginning of this section, we stated that the events happening are represented as raw events in data sources, restricted to information systems, see Definition 1. A data source is defined by its concrete location in the information system landscape. As shown in Figure 4 an event could be represented by none but also several raw events. In accordance with the motivational example described in Section 2, the raw event `<103, 2012-09-10 22:14:13, green>` captured in the database was created when Bob pushed the green voting button, see Table 1.

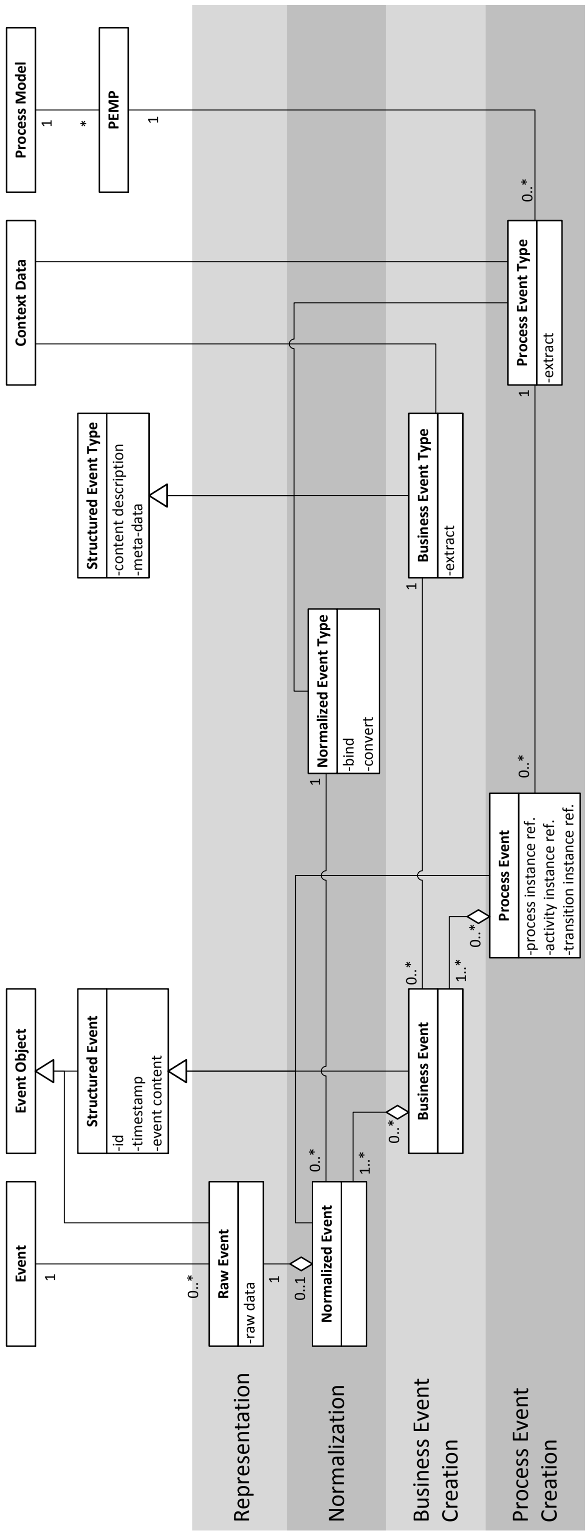

Figure 4. Event object processing phases towards process events

In contrast, the start of a movie is not represented by an raw event, because in the cinema's information systems is no information left about when the projector starts to show the movie itself.

3.2.2 Normalization

Raw events need to be extracted from the data sources for further processing. To set up the algorithms for enriching the event content, a unified base is required. Therefore, normalized events are introduced. A normalized event is a unified abstraction of raw events containing the raw data of the raw event in the event content. One raw event is normalized with exactly one event object, namely its normalized event, cf. Figure 4. However, some raw events are not normalized as they are not of interest for process intelligence.

Definition 3 (Normalized Event).
A normalized event is a tuple $E_N = (type, id, timestamp, C)$, where *type* is the normalized event type ET_N and *id* a unique identifier. *timestamp* is the point in time the normalized event occurred and C is event content representing the raw data of the corresponding raw event in a structured form. ◇

This means a raw event needs to be transformed into a normalized event by (a) storing the timestamp information when the raw event occurred in the timestamp attribute of the normalized event, and (b) storing the raw data of the raw event in a structured form, such as key-value-pairs, in the event content of the normalized event. These transformation rules are defined in the underlying normalized event type. In case, the raw event does not hold any information about the time the event happened, the timestamp of the resulting normalized events have to be derived somehow, e.g., by setting the timestamp of the normalized event creation. Despite [11] states that an event object could have several timestamps, e.g., for start and end time, we define the timestamp of an event object as the time it occurs. Aggregation of several raw events to one normalized event is not allowed in this phase.

Every normalized event is considered to be an instance of a normalized event type. The normalized event type holds a description of the event content and a function *bind* that defines how the required data are extracted from the data sources. Optionally, it can hold meta-data that describes the normalized events itself, e.g., for managing the types in a repository. With the optional conversion rules certain values of raw event data can be harmonized, e.g., time information from different time zones, units or currency. For instance, this could be required as normalized events are extracted from different servers in different time zones. The step of harmonization is done in this phase, because at this stage the properties of the data source, i.e., time zone, is known whereas in the further processing steps this information may not be available anymore, unless it is contained in the normalized event.

Definition 4 (Normalized Event Type).
A normalized event type $ET_N = (cd, bind, M, K)$ consists of a specification of the event content structure *cd*. Additionally, it holds the binding *bind* that points to the concrete data source. Optionally, it contains a set of meta-data M and and a set of conversion rules K. ◇

The normalized event type definition is in line with the event type definition described in [5] but applied to normalized events concretely.

While defining a normalized event type, an implementation must be given that indicates how the required information of the raw event could be extracted to form the normalized event.

Definition 5 (Binding, Implementation).
Let $\mathcal{ET}_\mathcal{N}$ be the set of defined normalized event types ET_N. A binding is a function $bind : \mathcal{ET}_\mathcal{N} \rightarrow I$ assigning an implementation to a set of normalized events, where I is the set of implementations, i.e., rules and methods to extract raw event information from the data sources within the information system landscape. ◇

Definition 5 allows to extract information about raw events in several ways, e.g., as a database query, as a service request, as a calculation method, as a stream processing filter, or as reading a log entry. These methods could be applied to several data sources where the raw event information is stored.

Referring to our running example, the definition of a normalized event type *voting* looks as follows.

```
voting.EventContentDescription = {ID,Rating}
voting.bind = {
   SELECT Timestamp, ID, Rating
   FROM rating.db
   INTO timestamp, ID, Rating;
   WHERE <condition>}
voting.metadata = {}
voting.conversion = {}
```

Listing 1.1. Definition of the normalized event type *voting*

Referring to our introduced example, the normalization of the raw event `<103, 2012-09-10 22:14:13, green>` will result in the following normalized event:

E_{N1}`=(voting,Y7E22,2012-09-10 22:14:13,{ID=103,Rating=green})`

In this case, the normalized event has the generated ID Y7E22. The timestamp is taken as it was given by the raw event. The raw data about the data record ID and the Rating are taken into the event content of the normalized event.

With the normalization, we get a unified base for further processing and can build up business events in the next processing step.

3.2.3 Business Event Creation

Business events combine several available normalized events and other business events and add business context data, see Figure 4. The combination of business events to another business event is unembodied in Figure 4, because all business events can be traced back to the utilization of normalized events. Here, the form of the normalized event is kept for business events but the event content structure is enhanced.

Definition 6 (Business Event).
A business event is a tuple $E_B = (type, id, timestamp, C)$, where *type* is the corresponding business event type ET_B and *id* a unique identifier. *timestamp* is the point in time the business event occurred and C is the structured event content. ◇

Creating a business event requires aggregation and composition of normalized events and business events as well as the utilization of additional data. Thus, several normalized events and business events can be combined to one business event. Aggregation in the phase of business event creation means that several normalized events of the same normalized event type are bundled to one business event. For example, in the cinema SuperMovies a specific business event is triggered while aggregating the corresponding normalized events if 90 % of the visitors pushed the red button while rating a certain movie. The created business event has significant influence on the program planning process, thus cancelling further planned movie shows could be an implemented reaction, for instance. Applying composition to form a business event means to combine several normalized events of different types. For instance, having a normalized event representing the switched off lights and seeing a normalized event for a voting (assuming only one movie is showing a time) could be composed to a business event indicating that someone leaves the movie earlier.

This means that (a) a timestamp needs to be set, and (b) all required and for the business event relevant information, specified by the business event type, needs to be stored in the event content in a strucutred form. Losing information of the normalized events and other business events while building up the event content of the business events may be intended in this phase, see projection in Section 3.1.

To describe how a business event looks like, a business event type has to be defined. Every business event is considered to be an instance of a business event type. This business event type definition describes how the timestamp is determined as well as how the event content is structured and where this data comes from. Analogously, to the normalized event type, a business event type definition contains a description of the optional meta-data.

Definition 7 (Business Event Type).
A business event type $ET_B = (cd, X_B, M)$ consists of a description *cd* that defines the structure of the event content. Additionally, it holds a set of extraction rules X_B that indicate which and how normalized events and other business events are utilized to form the business event as well as how additional context data could enrich the event content. Optionally, it contains a set of meta-data M. ◇

The business event type definition is in line with the event type definition described in [5] but applied to business events concretely.

Referring to our running example the definition of a business event type *movie_ voting* looks as follows.

```
movie_voting.EventContentDescription = {Rating,Movie}
movie_voting.extract = {timestamp=voting.timestamp;
    Rating=voting.Rating; Movie=programCW37.xls-Movie, where
voting.timestamp > programCW37.xls-Start+programCW37.xls-Duration AND
voting.timestamp < programCW37.xls-Start+programCW37.xls-Duration+15}
movie_voting.metadata = {}
```

Listing 1.2. Definition of the business event type *movie_ voting*

In the example of business event type *movie_ voting*, the information about the movie title is retrieved from the cinema program and added to the normalized event information. The other information contained in the cinema program are not of interest for the business intend. The data is correlated by time of the voting and the end of a movie.

Referring to our example, our previous introduced normalized event example will be enriched by the movie title “Rambo X”, cf. enrichment in Section 3.1, because the normalized event of the rating falls into the time period which last from the end of a movie plus 15 minutes. The resulting business event will look as follows:

E_{B1}`=(movie_voting,74WB4,2012-09-10 22:14:13,`
`{Rating=green,Movie=Rambo X})`

In this case, the business event got the generated ID 74WB4 and the timestamp is taken from the underlying normalized event. Since the original database ID 103 is not of interest anymore, it is left out in the business event type definition and thus not contained in the resulting business event, cf. projection in Section 3.1.

At this stage of event processing, an issue of correlating the right data with the corresponding event could occur, depending on the extraction rules defined. For example, it is not ensured that only visitors that leave a cinema hall after a certain movie ends do the voting. In our example, there is only one voting system installed in the whole cinema. So visitors that leave another movie earlier may vote also, but in context of another movie. Depending on the timeframe they vote, their voting could be correlated to another movie, e.g., someone leaves “Bad Boys 3” (movie show on September 10th, 2012 at 20:45) 10 minutes earlier, his or her voting will be correlated to “Rambo X”.

While building up business events, several normalized events may result in one business event. This could deliver significant value, especially if a real-world event was represented in several raw events that results in several normalized events. Bringing this information together represents the real-world event much better than the single raw events resp. normalized events do. For instance, assuming there are voting buttons installed at every seat and in a second database it is logged which seat voted already to avoid double voting. Thus, in the first data source, namely the voting results, cf. Table 1, is a representation of a visitor’s voting, but the database record in the second database, where only the occurrence of the voting per seat is logged, also. Bringing both representations together will result in a business event indicating the voting result and the corresponding seat number. This allows the analysis whether the voting is influenced by viewing direction of the visitor.

Once business events are built up, they could be used for monitoring and analysis. Some analysis requirements do not need the correlation to a concrete process instance. For instance, based on the number of votings and the number of sold tickets a business event could be created once the voter participation is not sufficient enough to have a good decision base, e.g., for keeping the movie in program or cancelling it. Bringing business events into a process context will add further information to the event content and allows process intelligence techniques.

3.2.4 Process Event Creation

Process events are described by a process event type. The procedure of setting up a process event type is split in a design and a configuration phase. During the design, the position when a process event is expected is defined within the process model. The so-called process event monitoring point (PEMP) is a tuple of the process model it belongs to, the corresponding node, and its state transition. This is the same as it is done for event monitoring points described in our previous work [8].

Stating at which position of the process which process event is expected on process node level is too coarse-grained. Thus, applying life cycle models to the process nodes allows a much more fine-grained positioning to the state transitions of each node. For each of the nodes, we envision a state-based life cycle model, where we reserve the flexibility, to assign a unique life cycle model to any node. Life cycles of process nodes have been exhaustively discussed in literature [23,20]. We employ a generic life cycle model.

Definition 8 (Life Cycle Model).
A life cycle model $L = (S, T)$ consists of states S and state transitions $T \subseteq S \times S$ being the transition from one state to another. $\mathcal{L}$ is the set of life cycle models.

Let $P = (N, F)$ be a process model. There exists a function $\varphi : N \rightarrow \mathcal{L}$ that assigns a life cycle model to every node $n \in N$ of P. $\diamond$

State transitions are the elementary facts that can be leveraged to position a PEMP. The set of all state transitions of a process model $P = (N, F)$ is comprised by $\bigcup_{n \in N} \{(n, t) | t \in T_{\varphi(n)}\}$, each of which could be potentially linked to a PEMP. $T_{\varphi(n)}$ represents all transitions $t \in T$ returned by function $\varphi(n) = (S, T)$ for node $n \in N$.

Definition 9 (Process Event Monitoring Point).
Let $P = (N, F)$ be a process model and $L = (S, T)$ be a life cycle model. A Process Event Monitoring Point is a tuple $PEMP = (P, n, t)$, where $n \in N$ is the node it is created for, and the state transition $t \in T_{\varphi(n)}$ within the life cycle of node n it is assigned to. $\diamond$

The PEMP represents the position in the process model where a process event can be correlated to.

Definition 10 (Process Event).
The process event $E_P = (type, id, timestamp, p_i, n_i, t_i, C)$ is a tuple, where *type* is the corresponding process event type ET_P and *id* a unique identifier. *timestamp* is the point in time at which the event is occurred. p_i holds the reference to the corresponding process instance, n_i the node instance, and t_i the instance of the transition analogously. C is the event content. ⋄

Every process event is considered to be an instance of an process event type. The process event type contains the specification of the event content that is required in this process context. The process event type holds a reference to a PEMP to include the position at which a process event of the particular process event type is expected. During configuration it is defined how the timestamp and the event content of the process event is created.

Definition 11 (Process Event Type).
A process event type $ET_P = (cd, PEMP, X_P, M)$ consists of a description *cd* of the event content structure. It holds the reference to a process event monitoring point $PEMP$ it is assigned to. Additionally, it holds extraction rules X_P that indicate which and how business events and other process events are utilized to form the process event as well as how additional context data could enrich the event content. Optionally, it contains a set of meta-data M. ⋄

The rules for the correlation of the process event to a process instance are described in the process event type as well and are part of the extraction rules X_P. The process event type definition is in line with the event type definition described in [5] but applied to process events concretely.

During the **process event type design** phase, a process event type is created, and placed in a process model, i.e., bound to a PEMP. The definition of process event types is mainly driven by process intelligence goals that may be influenced by legal regulations and best practices.

Referring to the cinema example, one may be interested in the end of the activity *Show Commercial* and in the end of the activity *Gather Feedback*. Therefore, the corresponding process model "Show Movie", cf. Figure 1, is used to define the process event types *Show_ Commercial_ End* (PET_1) and *Gather_ Feedback_ End* (PET_2), cf. Figure 5. For simplicity reasons, the activity life cycle, consisting of states *enabled*, *running* and *terminated* with the state transitions *(e)nable*, *(b)egin* and *(t)erminate*, is assigned to every relevant activity in the process model.

After creating and assigning a process event type, the data of interest (event content) is defined. Thus, at the end of the design phase, the process event type is created, a PEMP is assigned, and has a defined event content structure.

Returning to our example, the process event representing the end of the activity *Gather Feedback* will look as follows:

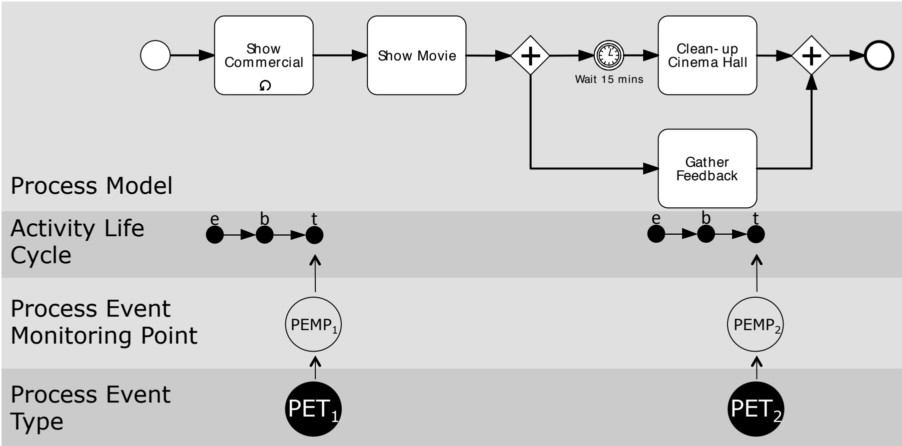

Figure 5. Process with defined and placed process event monitoring points

```
Gather_Feedback_End.EventContentDescription = {}
Gather_Feedback_End.pemp = PEMP2
Gather_Feedback_End.metadata = {}
```

Listing 1.3. Definition of the process event type *Gather_Feedback_End*

Note, that the process event type *Gather_Feedback_End* is presented as PET_2 in Figure 5.

In this example we do not need any event content, because we are only interested when the activity *Gather Feedback* ends for a certain process instance. This information could be derived from the mandatory timestamp attribute of the process events. In the process event type configuration step it is defined which data is required and the rules about the data processing are specified.

In the **process event type configuration** it is defined which event data, namely business events and other process events, is used, how it is correlated to a certain process instance, and how context data needs to be utilized.

Regarding our example, the business events that are of interest are those, which relate to a certain movie (movie_voting.Movie) and lie in a timeframe that corresponds to the process instance. Out of this set, the latest business event is of interest. For simplicity reasons, we summarize those conditions as *correlation_rules* in the following implementation of the extraction rules X_P. *Gather Feedback* will look as follows:

```
Gather_Feedback_End.extract = (timestamp = movie_voting.timestamp,
    where {correlation_rules})
```

Listing 1.4. Definition of the process event type *Gather_Feedback_End*

With this process event type definition, the required process events could be created. For example, the process event *Gather_Feedback_End* for the process instance of showing the movie “Rambo X” on September 10th, 2012 at 20:15 will look as follows:

E_{P1}=`{Gather_Feedback_End,41T70,2012-09-10 22:16:55,`
`Rambo X on September 10th, 2012 at 20:15,`
`Gather Feedback(1),terminate,{}}`

In this case, the process event got the generated ID 41T70. The process event is related to activity instance 1 of activity *Gather Feedback*. The underlying raw event information that is necessary to build this process event is coming from the rating database rating.db with id 104 (latest one that is lying in the timeframe of 15 minutes after the movie show of “Rambo X” ended).

3.3 Process Event Consumption

To complete this section, some process event consumer should be named exemplarily. Besides the existing business events, process events are available for process intelligence applications such as process monitoring and analysis. Process events can be used to calculate key performance indicators (KPI) that require process context, e.g., process cost calculations, but also allow the display of process execution data directly in the corresponding process models. A log of process events could build a reliable source for process mining techniques listed in [22].

3.4 Summary

In this section, we introduced the event objects raw event, normalized event, business event, and process event that base on each other, see Figure 6. Every event object is described by an event object type that indicates among others how the corresponding event object has to be created. These event objects and event object types are subject of several event object processing phases, namely representation, normalization, business event creation, and process event creation, see Figure 6.

Phase	Type	Event	Relation
Process Event Creation (Section 3.2.4)	Process Event Type (Definition 11)	Process Event (Definition 10)	
Business Event Creation (Section 3.2.3)	Business Event Type (Definition 7)	Business Event (Definition 6)	
Normalization (Section 3.2.2)	Normalized Event Type (Definition 4)	Normalized Event (Definition 3)	
Representation (Section 3.2.1)		Raw Event (Definition 1)	

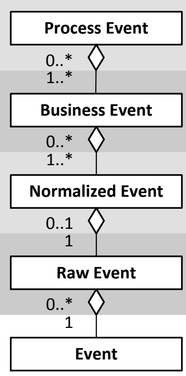

Figure 6. Overview of the phases, event object types and event objects including their relationships

4 Application to Running Example

Referring to the motivating example described in Section 2, the approach is applied to the data, events and processes of the cinema SuperMovies.

As *raw events* in the cinema, we refer to the data about the voting results and the lighting status. A raw event is created in the database when Bob pushes the green button because he likes the movie "Rambo X", e.g., `<103,2012-09-10 22:14:13, green>` . The *normalization* of this raw event will result in the following *normalized event*:

E_{N1}`=(voting,Y7E22,2012-09-10 22:14:13,{ID=103,Rating=green})`

Based on this normalized event, we can start to add business context data. By including context data, a business context can be added to the normalized event. In the cinema, the context data is represented by the cinema program that exists in parallel to the data available for every single process instance of a movie show. Applying the program information to the normalized event mentioned above, we can add the information of the movie rated:

E_{B1}`=(movie_voting,74WB4,2012-09-10 22:14:13,`

`{Rating=green,Movie=Rambo X})`

Business Events are valuable for monitoring and analysis questions already. Nonetheless, adding process context information could bring in even more insights, e.g., regarding the process progress. In the cinema, we can derive from the voting when the activity *Gather Feedback* terminates, e.g.:

E_{P1}`={Gather_Feedback_End,41T70,2012-09-10 22:16:55,`

`Rambo X on September 10th, 2012 at 20:15,`

`Gather Feedback(1),terminate,{}}`

This information is useful for evaluating the process run time for improvement reasons.

As shown, the focused information provides less value, but combing those gains more information content. For example, enriching the normalized events for the voting results with the movie name and building up the corresponding business event allows for concrete movie related analysis, e.g., all ratings for "Rambo X". Adding process context to single event information allows for monitoring process progress and evaluation of it, e.g., how long does the rating of a movie usually take. This information could be used for adjusting the deadline until when the voting results are assigned to a movie (in this paper we used 15 minutes). With the correlation of an event object to a process instance, we meet the requirements for a 4-star event object, cf. Table 3. Thus, we raised the quality of the original raw event by applying our approach.

To summarize this section, we can conclude that implementing the approach in the cinema, the originally separated data (cf. Figure 2) events, context and processes are interconnected and deliver more insights into the process execution, see Figure 7.

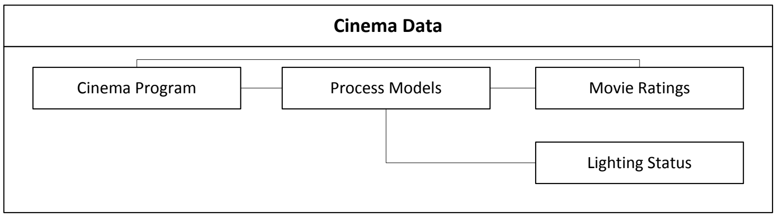

Figure 7. Data in a cinema and their interconnections after applying the proposed approach

5 Challenges

The presented approach involves several challenges that have to be researched, solved, and engineered in future work to get a highly flexible data source for process intelligence. We categorize the challenges to be solved according the phases of event object processing, we defined in Section 3 starting with challenges that targets to all phases, and describe the challenges around the whole approach in Section 5.7. Since *process event creation* and *business event creation* are the major subject of our future research we will consider them first. A summarized overview over all challenges identified so far is given in Section 5.8.

5.1 General Challenges

There are four challenges that targets to all phases. Thus, these will be discussed first. First of all, there need to be an authorization concept in place (GE-1). This is necessary, because there may are events in the system that are highly confidential an should be provided only to a couple of people, e.g., the worker should not get access to events that are relevant and directed to C-level management.

The available event objects need to be easily found for the usage in the phase build-on certain event objects resp. by the end consumer. Therefore event object type repositories could be used that holds the event object type definitions but provide search capabilities by text and other criteria also (GE-2). Additional to the organization of event object types in repositories, a grouping resp. categorization/clustering of them could be useful for supporting (semi-)automatic configuration for example (GE-3). We expect that the structured event types evolve over time. This may requires versioning of structured event types and version numbers in the single structured events, to indicate under which definition they were created (GE-4). For instance, in version two of the business event type *movie_voting* the event content is enhanced by a field for the producer. However, the business events created under the specification of business event type version one will not contain this information.

5.2 Process Event Creation

Based on the available business events, process events can be build. Additional to the business event content, context data may be of interest as well. How could an implementation for process event creation look like that allows the combination of those (PE-1). Which existing techniques and methodologies, e.g., query languages, could be utilized within that phase. When we are talking about process intelligence, it is important to know what needs to be monitored and analyzed. Therefore, event monitoring models could be build up that describes what measurements are mandatory, e.g., by law or standards, which are nice to have and valuable, and which are irrelevant (PE-2). Building up the event monitoring models requires a lot of domain knowledge. Having a model for monitoring and analysis provides the ability to do conformance checking according to guidelines, laws, or standards.

Process event creation may depend on certain master data or specific domain knowledge that is important to be incorporated (PE-3). For instance, voting results should ny considered only when visitors are 50 years old in average.

While defining process events, we gain a lot of information by utilizing the process model the process event is defined for. Around that area a lot of challenges need to be solved and several ideas need to be shaped. A very initial challenge is the technical implementation of process event type definition in the process model directly (PE-4). How could that information be further used and leveraged for instance for design-time analysis (PE-5)? While defining process event types a causality between these is established through the process model. How could this information be used during design-time as well as in process intelligence (PE-6). Focusing on modeled activities it is interesting how model context information, such as lanes and resources, are relevant for process intelligence (PE-7). One can think of using data object state change information to represent an event (PE-8). During runtime an activity could be performed multiple times. Thus, a concept for monitoring multiple activity instances need to be invented (PE-9). Currently, activity monitoring is focused, but monitoring the control flow between two activities, namely the flow edges, is of interest too, e.g., if the flow represents a change in process location that consumes time (PE-10).

A huge challenge in business process management is the handling of process evolution, namely the change of process models over time. This applies for process events and their usage in process intelligence applications too. Central question at this point is whether process events are comparable to each other or not, because they may captured in an evolved process (PE-11). Initial work in that area was done in a master thesis by Maschke [13] that need to be enhanced. Besides the process model itself, meta-data of it can support in defining process events as well (PE-12). For instance, the process description may be used for deriving process events. It needs to be investigated further whether an process event could always point to a specific point, namely a state change in the activity. Process events are maybe relevant for a whole region in the process, e.g., a group of activities (PE-13). The approach presented follow a bottom-up approach, starting with the raw event to form business events and later on process events. But how can we benefit from business events that are not utilized in process events yet, but relevant for the process (PE-14). This situation could occur because of just not utilizing this event information, but also because an event indicating an exception. Another topic that need to be investigated is about data cleansing. How could the event information be prepared to suppress certain side-effects (PE-15). This can be done by using context data, e.g., using a calender for suppressing the cinema's day of rest, e.g., Mondays, from event information in a production process.

5.3 Business Event Creation

When using normalized events to form business events a lot of challenges need to be tackled. One interesting question is, how the event content of a business event is defined and how the utilization of normalized events and other business events look like. Therefore, certain techniques and methods need to discovered and developed, such as different query languages (BE-1). One can think of using an existing event taxonomy, e.g., domain-specific taxonomies, to define the event content structure of a business event automatically. While utilizing business events for analysis, it is interesting to cluster business events by certain criteria, that need to be defined beforehand (BE-2). Similar to process events, a solution needs to be provided to benefit from raw resp. normalized events that are relevant for my business but not utilized in any business event definition (BE-3). Almost identical to the data cleansing challenge for process events, business event information must be processed accordingly (BE-4). Once the business events are defined, they could be provided for monitoring and analysis applications (BE-5) but also to form process events.

When correlating several information of a normalized event to a business event special intention has to be spend on the process execution in different systems (BE-6). It may occur that in different systems the steps performed relating to one process instance may be identified by different identifiers (ID).

5.4 Event Normalization

Within the normalization step, a lot of research has been conducted already, that is describing how an event could be structured etc. Therefore, an intensive literature review is necessary (NO-1). In this phase, the event information represented as raw events needs to be accessed. Thus, techniques and methods need to be defined to provide the necessary data access in an easy to configure, high-performing and secure way (NO-2). There may are possibilities to support the normalization of raw events by using certain semi-automated or even full-automated approaches (NO-3).

5.5 Event Representation

In this area, a lot of literature is available already. The main challenge in this phase is to summarize the literature and get a complete picture about techniques, methods, and frameworks that provide solutions in that area (RE-1). Especially event log standards have to be explored (RE-2). Generating more process execution information through additional event representations is not in scope of this document. Thus, further research could be done about the techniques and methods on how more events, occurring during manual executed processes, could be represented without increasing the effort for the human processor (RE-3). In this area, techniques are required that captures events without any human interaction, e.g., RFID. Since information systems are moving into a cloud environment more and more, it is necessary to have an answer on the question how the events occurring in the cloud could be captured, assessed, and represented as raw event (RE-4).

A more domain-specific problem is the deferred representation of occurring events (RE-5). This is the case in health care for example. For example, during ward rounds nowadays data and information are still captured by writing them down to paper first before they are entered to the clinical information system later on. The event itself happened during the ward round, e.g., medication, but the representation of this event in the information system landscape is available to a later point in time.

5.6 Process Event Consumption

For completeness reasons, we will list a couple of challenges that relates to the process event consumption and their usage. Initially, it has to be clarified how process events can be utilized in process intelligence applications (PC-1). We refer to a process cockpit as the umbrella for such applications. The idea behind a process cockpit is that all required information about the process and the process executions are accessible in one place (PC-2). Therefore, it is essential to discuss what is relevant in such a cockpit and how the single views have to look like. This is a huge topic that needs to be evaluated with end users (PC-3). In this area it is vital to simplify the build of process cockpit views, e.g., by pattern templates for certain analytical questions (PC-4) or by (semi-) automatic creation of process cockpit views based on the underlying process model. At this stage, it may helps to collect typical monitoring and analytical questions and categorize them (PC-5). During realization of process intelligence it has to be differentiated between process instance and process measures (PC-6). As learnt from implementation projects it is also required to differentiate monitoring approaches per process instance, e.g., a process relating to customer A needs to be updated every ten minutes, the same process relating to customer B needs to be updated only at significant process progress steps (PC-7).

One key aspect in process cockpits will be key performance indicators (KPI) for sure. So it is essential to investigate on KPI and measurement modeling (PC-8). How are KPIs handled that are influenced by several processes not just one? An interesting question that arises when introducing KPIs into process cockpits is at which depth they are related to PEMP. One can think of the generation of PEMPs out of KPIs or a design-time analysis whether a KPI could be answered by the available process events.

5.7 Relating Approach

For the whole approach a guideline needs to be created that indicates besides a step-by-step description also a role concept to describe who is doing what (GE-5).

The guideline could be supported by the presentation of the approach in the Camunda BPMN-Framework [6].

It is necessary to propose techniques and methods for automating several steps to a certain extent (GE-6). Currently, the setup of the approach requires a lot of manual work which could supported by (semi-) automatic configuration and definition steps.

The proposed approach is relying on the quality of data in the underlying IT system landscape. Thus, an intensive discussion about that is necessary to show the boundaries and limits of the approach (GE-7).

Once all the challenges mentioned above are solved and answered, the approach could be lifted to the next level, namely working in a process architecture (GE-8). In that area, especially the relations between the process events of different processes are of interest.

As a parallel step, we will also look into the area of adaptive case management and how the approach could be applied in such environments (GE-9).

5.8 Challenge Overview

In Table 5 we list all the challenges identified so far.

Process Event Creation (PE-1 to PE-15)	
PE-1	Implementation of process event creation
PE-2	Event monitoring models
PE-3	Process event creation dependent from master data resp. domain knowledge
PE-4	Definition of process event types in the process model
PE-5	Process event type definition design time analysis
PE-6	Derivation of causality of process events based on process models
PE-7	Defining process events based on activity context such as resources and lanes
PE-8	Process events based on data object state changes
PE-9	Process event creation in the context of multi-instance activities
PE-10	Process events during control flow
PE-11	Influence on process events caused by process evolution
PE-12	Are process model meta-data, such as process description, valuable for process event type definition?
PE-13	Process events related to a region of process nodes
PE-14	Using valuable structured events that are not utilized for process event creation
PE-15	Cleansing of process events

Business Event Creation (BE-1 to BE-6)	
BE-1	Implementation of business event creation
BE-2	Grouping of business events with respect to the organization's point of view
BE-3	Using valuable structured events that are not utilized for business event creation
BE-4	Cleansing of business events
BE-5	Using business events for monitoring and analysis
BE-6	Correlation of normalized events to form business events
Event Normalization (NO-1 to NO-3)	
NO-1	Literature review
NO-2	Implementation of normalized event creation
NO-3	(Semi-) automated approaches for normalized event creation
Event Representation (RE-1 to RE-5)	
RE-1	Literature review
RE-2	Evaluation of log standards
RE-3	Approaches to represent events
RE-4	Event representation in the cloud
RE-5	Deferred recording resp. representation of events
Process Event Consumption (PC-1 to PC-8)	
PC-1	Process event utilization in process intelligence applications
PC-2	Process cockpit
PC-4	Process monitoring view templates
PC-5	Categorization of process analysis requirements
PC-6	Process and process instance measures
PC-7	Different process monitoring approaches per process instance
PC-8	Relationship between process resp. business events and KPIs
General Challenges (GE-1 to GE-4)	
GE-1	Authorization concept for structured events
GE-2	Repositories for structured event types
GE-3	Structured event type clustering/grouping to enable (semi-) automatic configuration approaches
GE-4	Versioning of structured event types and structured events
Relating Approach (GE-5 to GE-9)	
GE-5	Guideline about the usage of the framework including a role concept
GE-6	Automation of several steps in the framework
GE-7	Discussion of data quality and how it can be improved
GE-8	Adoption of the framework to process architectures
GE-9	Adoption of the framework to case management

Table 5. Challenge overview

6 Related Work

The presented approach is based on several event object processing steps that include the creation of new event objects based on available information, e.g., the creation of business events based on normalized events. These event object processing steps are heavily related to the topic of complex event processing (CEP). [11] gives a good introduction about CEP and introduces techniques and concepts of event processing, such as event patterns, rules, event pattern languages etc. These techniques and concepts could be applied in our approach to build up the structured events of the business event creation and process event creation phase. In [5], the authors introduce CEP as well and define certain CEP-related terms, such as event type, that are applied in the definitions found in our paper. Further terms in this paper are defined according to the definitions published by the event processing technical society [12]. The discussion of several challenges that are listed in this paper, e.g., the discussion about security, is initiated by [9].

One problem that has to be addressed when different event object sources for a process exist is correlation of event objects to one case resp. process instance. Motahari-Nezhad et al. [15] provide algorithms to determine correlation sets on different attributes of events for distributed environments. They use methods of atomic, conjunctive, and disjunctive correlation conditions and heuristics to find correlating groups. The aspects of correlation are also relevant for this paper with respect to the correlation of several event objects to each other but also to the process instance. However, the focus of this paper is on how to map correlated business events to a process event in a flexible architecture.

The approach presented in this paper aims to be a high-quality event basis for business process intelligence. The capability to monitor, visualize, and evaluate business process execution is one core topic of business process intelligence (BPI) [16], which addresses "managing process execution quality by providing several features, such as analysis, prediction, monitoring, control, and optimization" [7]. A load of research discusses the capturing and storing of process execution data for evaluation purposes [7,1,14], but most works assume the ability to log every process step to ensure a complete event log. In [16], the authors argue that process monitoring and analysis are vital to BPI and propose, based on the specific requirements of BPI, a reference architecture, composed of an integration layer, a functional layer, and a visualization layer. The approach presented in this paper targets at the integration and the functional layer, i.e., provides means to integrate data, process knowledge, and information about process execution and combine those. Rinderle-Ma et al. [19] discuss the need of process views to strengthen the understanding of a business process according to the users' needs. In the same vein, the requirements for a process monitoring system for system-spanning business processes are discussed and evaluated in [2]. Nevertheless, the visualization topic is not discussed in this paper in detail. The presented work can be seen as input for these techniques. However, some basic ideas how it could be implemented in form of a process cockpit are given.

Dahayanake et al. [3] give an overview of business activity monitoring (BAM) and introduce four classes of BAM systems: pure BAM, discovery-oriented BAM, simulation-oriented BAM, and reporting-oriented BAM. Since all of these classes base on so-called raw events, our presented approach could be applied as well to provide a high-quality basis of events, while the business resp. process events serve as input events for BAM techniques and methods.

With regards to business process evaluation, the concept of process performance indicators (PPI), the process related form of key performance indicators, is introduced in BPM. Del-Río-Ortega et al. [4] introduce an ontology to define PPIs for measuring process execution performance, such as time, costs, and occurrences. These PPIs can be applied directly on top of our approach, as it provides the base for measurements that can be compared to target values. As these measures can already be provided while the process instance is running, violations of tolerance thresholds can be mitigated before the process instance failed a PPI.

Process mining [21] is a discipline that can be used on top of the process events merged in an event log to extract all kinds of process information, e.g., execution times and conformance checks to existing process models. The main difference to the approach presented in this paper is that we utilize a top-down approach of connecting (detailed) process models to event objects, while process mining is a bottom-up approach based on logs.

7 Conclusion and Outlook

Business process management is a key topic for companies to be successful in a competitive market. Especially the improvement of their business processes is important for which a reliable data foundation is required to monitor and analyze the execution of business processes. This is not a trivial problem, because in especially manual process executing environments, data about the process execution that is linked to the process already is rather sparse. In this paper, we present an approach that utilizes as much information of the companies' information systems as possible to enable meaningful process intelligence. Not only event information created during process execution is subject of the proposed processing steps, but also context data and process knowledge. This information is included during event processing step by step, when (a) normalizing raw event data from several data sources, (b) creating business events out of that normalized events and adding business context data, and (c) creating process events by combining business events and process knowledge. The result of these event processing steps are high-quality process events that can be consumed by process intelligence applications, such as process monitoring, process analysis, and process mining.

In the future, we will tackle the challenges mentioned in Section 5 one by one focusing especially on process and business events. Nevertheless, real-world events, raw event, normalization and process event consumption will be considered as well to provide an end-to-end solution finally.

References

1. B. Azvine, Z. Cui, D.D. Nauck, and B. Majeed. Real Time Business Intelligence for the Adaptive Enterprise. *IEEE CEC/EEE'06*, pages 29–29, 2006.
2. R. Bobrik, M. Reichert, and T. Bauer. Requirements for the Visualization of System-Spanning Business Processes. In *BPMPM'05/DEXA'05*, pages 948–954. IEEE Computer Society Press, 2005.
3. A. Dahanayake, R.J. Welke, and G. Cavalheiro. Improving the understanding of BAM technology for real-time decision support. *Int. J. Bus. Inf. Syst.*, 7:1–26, December 2011.
4. A. Del-Río-Ortega, M. Resinas, and A. Ruiz-Cortés. Defining Process Performance Indicators: An Ontological Approach. *On the Move to Meaningful Internet Systems: OTM 2010*, pages 555–572, 2010.
5. O. Etzion and P. Niblett. *Event Processing in Action.* Manning Publications Co., 2010.
6. J. Freund and B. Rücker. *Praxishandbuch BPMN 2.0.* Carl Hanser Verlag München Wien, 2012.
7. D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M. Shan. Business Process Intelligence. *Computers in Industry*, 53(3):321–343, April 2004.
8. N. Herzberg, M. Kunze, and A. Rogge-Solti. Towards Process Evaluation in Non-automated Process Execution Environments. In *Proceedings of the 4th Central-European Workshop on Services and their Composition, ZEUS*, 2012.
9. A. Hinze and A. Buchmann. *Principles and Applications of Distributed Event-Based Systems.* Idea Group Reference, 2010.
10. G. Keller, M. Nüttgens, and A.W. Scheer. Semantische Prozessmodellierung auf der Grundlage "ereignisgesteuerter Prozessketten (EPK)". *Veröffentlichungen des Instituts für Wirtschaftsinformatik*, 89, 1992.
11. D. Luckham. *The Power of Events.* Addison-Wesley, 2002.
12. D Luckham and R. Schulte. Event Processing Glossary - Version 1.1, July 2008. `http://www.ep-ts.com/component/option,com_docman/task,doc_download/gid,66/Itemid,84/`.
13. P. Maschke. Process Evolution & Evaluation – Analyzing versioned execution data of clinical pathways, April 2012. Master Thesis.
14. F. Melchert, R. Winter, M. Klesse, and N.C. Jr. Romano. Aligning Process Automation and Business Intelligence to Support Corporate Performance Management. In *AMCIS'04*, pages 4053–4063, New York, 2004. Association for Information Systems.
15. Hamid Reza Motahari-Nezhad, Regis Saint-Paul, Fabio Casati, and Boualem Benatallah. Event correlation for process discovery from web service interaction logs. *The VLDB Journal*, 20(3):417–444, June 2011.
16. B. Mutschler and M. Reichert. Aktuelles Schlagwort: Business Process Intelligence. *EMISA Forum*, 26(1):27–31, January 2006.
17. OMG. Business Process Model and Notation (BPMN) 2.0 Specification, January 2011. `http://www.omg.org/spec/BPMN/2.0/PDF`.
18. M.E. Porter. *Competitive Advantage: Creating and Sustaining Superior Performance.* Free Press, 1998.
19. S. Rinderle, R. Bobrik, M. Reichert, and T. Bauer. Business Process Vizualization - Use Cases, Challenges, Solutions. pages 204–211, 2004.
20. N. Russell, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Workflow resource patterns: Identification, representation and tool support. In *Advanced Information Systems Engineering*, pages 216–232. Springer, 2005.

21. W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag New York Inc, 2011.
22. W.M.P. et. al van der Aalst. Process Mining Manifesto. In *7th International Workshop on Business Process Intelligence (BPI 2011)*, pages 169–194, Campus des Cézeaux, Clermont-Ferrand, 2012. Springer-Verlag.
23. M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag New York Inc, 2012.

Aktuelle Technische Berichte des Hasso-Plattner-Instituts

ISBN 978-3-86956-241-4
ISSN 1613-5652