

# Improving Process Monitoring and Progress Prediction with Data State Transition Events

Nico Herzberg and Andreas Meyer

Hasso Plattner Institute at the University of Potsdam  
{nico.herzberg, andreas.meyer}@hpi.uni-potsdam.de

**Abstract.** Monitoring business processes during their execution is one important aspect of business process management. Process monitoring requires observed events, which are recorded in information systems, to reason about, amongst others, process progress. Especially in manual executing process environments, the observed events are most likely sparse. Therefore, we introduce an approach increasing the number of observed events by capturing data state transition events, which occur after successfully writing a data object during process execution.

## 1 Introduction

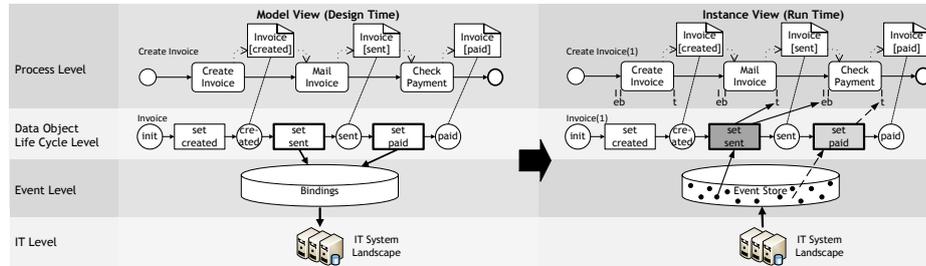
In the field of business process management, monitoring is used to observe process behavior and probably to react upon events as well as to predict upcoming process steps during process execution. Processes automated by information systems, i.e., process engines, can be monitored very well because the system usually provides logging capabilities and therefore, the progress is easily recognizable. In contrast, in environments with processes to be mainly executed manually, such as in health care, many occurring events are not captured. Thus, event information about such processes is incomplete and exact proposition about progress, e.g., by applying the concept of event monitoring points [1], is not possible. An event monitoring point is correlated to certain events captured by an IT system connected to a specific event data source, e.g., a database or a bar code scanner, and informs when certain states transitions as, for instance, enable, begin, or terminate, occur in a process activity. Probabilistic means, as for instance explained in [4], can provide an indication about process progress but are an approximation only.

Nevertheless, progress recognition of the complete process and an holistic view on the process are goals in manual executing process environments also. Introducing additional documentation and logging activities during process execution can be a solution, however, this is not intended, because it creates additional effort to the process participants, e.g., the doctors and nurses. Therefore, we will introduce an approach utilizing events from data object creation or modification to increase the number of observable events used for process monitoring and prediction of process progress. We call them *data state transition events*. After the write of a data object in a specific data state, we can assume this data object to be existent. Besides arguing about process progress, the registration of

events helps in deciding about future process execution and the chance of proper completion of a process or the reachability of a specific activity. Additionally, the approach can be utilized to identify incorrect behavior similarly to what is done in the field of data conformance, e.g., [3].

## 2 Approach

The presented approach enables insights into process execution based on information about a data object. Data objects and their life cycles are the basis for this approach. A data object life cycle (OLC) is represented by a Petri net, where a place describes a data state and a transition represents a data state change from the preceding to the succeeding one. An OLC specifies all allowed data state changes of the corresponding data object. As an example, we refer to an *Invoice* with data states *created*, *sent* and *paid*, whereat these states can be reached in that order only, see Fig. 1 – Data Object Life Cycle Level. Based on this OLC, those transitions are selected that can be monitored with events. We refer to an event as a happening in a particular *point in time* at a certain *place* in a certain *context* that is represented in the IT system landscape, see Fig. 1 – IT Level. The observable data state transitions of the OLC have to be linked to the events that provide the information about the triggering of the transitions. This link is provided by a binding, see Fig. 1 – Event Level, of the particular transition of the OLC to an implementation that extracts the necessary information about the events from a data source including the correlation to a specific data object instance during runtime. For the binding, we use the same technique as presented in our earlier work [1]. In our example, we assume that sending the invoice and receiving the payment can be observed by retrieving the relevant data from a database.



**Fig. 1.** Scenario describing the approach at design time and run time

During the design of several process models, the data objects can be assigned to particular activities, whereas the activities either read a data object in a certain data state, create a data object in a certain data state, or transfer a data object from one data state to another. We assume that the usage of the data object in a process correlates correctly to the OLC, i.e., the process only uses the specified data states and their transitions as shown in Fig. 1 – Process

Level. Nevertheless, not every possible data object state in the OLC needs to be reflected in the process model.

Based on the assignment of the data objects to the activities, it is possible to provide information about the process execution during runtime. We assume that an activity is enabled if the activity could be executed because of the control flow specification and if the input data object is available in the required data state. Furthermore, we assume that an activity is terminated as soon as the output data object is present in a certain data state. Information about process execution can be provided for activities, which consume or provide data objects in certain data states whose state transition is observable.

Referring to our example, the termination of activity *Create Invoice* and the enablement of *Mail Invoice* cannot be monitored with information about the data object *Invoice* only, whereas the completion of the second activity as well as the enablement of the third activity of the process model can be tracked on IT Level. This is visualized by the edges to the *Bindings* database in Fig. 1 (left part). At design time, the data state transitions to *sent* and *paid* are marked as observable (bold outline). This maps to the aforementioned monitoring capabilities at run time. At run time, the progressing in the process can be recognized by data state transition events occurring in the event store. In Fig. 1 (right part), an event for sending *Invoice(1)* (data object with corresponding instance id) is observed in the event store (solid edge), but there is no event available for the payment although it is expected to happen some time (dashed line). For monitoring process details not observable by the means of data manipulation, the introduced approach can be combined with other approaches, e.g., [1].

For simplicity reasons, we described the approach assuming at most one input data object and at most one output data object per activity. However, the approach is also valid for many input and output data objects.

### 3 Related Work

In [1], a framework is presented that describes the definition of so-called event monitoring points in business process models based on their activities' states. This work builds the basis of the approach discussed in this paper. As process mining techniques [5], especially conformance checking, are relying on event logs that are complete with respect to the underlying process model, the presented approach could help to enrich existing event logs especially in manual executing process environments and extend them to complete event logs. [3] checks for conformance between process models and data objects at design time while [2] ensures compliance to regulations by restricting the design of artifact-centric, i.e., data-centric, processes. In contrast to these works, our approach targets on run time compliance to data objects.

### 4 Conclusion

The presented approach uses information about data objects and their data states resp. data state transitions from process models to enable process monitoring.

The observable data about the transition of a data object from one data state to another provides further insights about the process execution. Thus, a more detailed process monitoring can be assured.

## References

1. N. Herzberg, M. Kunze, and A. Rogge-Solti. Towards Process Evaluation in Non-automated Process Execution Environments. In *Services and their Composition (ZEUS)*, 2012.
2. N. Lohmann. Compliance by design for artifact-centric business processes. In *Business Process Management*, pages 99–115. Springer, 2011.
3. A. Meyer, A. Polyvyanyy, and M. Weske. Weak Conformance of Process Models with respect to Data Objects. In *Services and their Composition (ZEUS)*, 2012.
4. A. Rogge-Solti and M. Weske. Enabling Probabilistic Process Monitoring in Non-automated Environments. In *BMMDS/EMMSAD*, pages 226–240, 2012.
5. W. M. P. van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.