



Multi-Event Log in Celonis Process Mining 4.7

January 2021

Table of Contents

Table of Contents	2
Motivation for Multi-Event Log	3
Limitations of traditional Process Mining	3
Opportunities with Multi-Event Log Process Mining	4
Multi-Event Log in a nutshell	5
Set up of a Multi-Event Log Data Model	6
Data Transformation	6
Data Model	10
Set up of a Multi-Event Log Analysis	12
Discover multiple processes	12
Configure Process Visualization in Process Explorer and Variant Explorer	12
Create KPI Calculations across processes	14
Metrics from different Event Logs in one Analysis	14
Filter across multiple processes	15
Metrics across different Event Logs	15
Discover parallel processes	16
Limitations of the current Analysis regarding Multi-Event Log usage	18

Motivation for Multi-Event Log

Questions you can answer with Process Mining:

- How much time do you need to sell a product after it was ordered from you?
- If you realise you don't have this product in stock but have to buy it from another vendor, how long does it take until it arrives in your warehouse?
- Are you paying the vendor according to the best possible payment terms?

In the Process Mining world so far, these are three different perspectives on the data that would require three separate Data Models (see below), three different Analyses and a lot of mental overhead to analyze those. Discover in the following how you can combine the three perspectives into one with the new Multi-Eventlog capability.

Limitations of traditional Process Mining

Traditional Process Mining follows one record type through the system. For this record type, all events related to the record are collected in an activity table and linked together with a case ID.

In general an entry in an Event Log consists of:

- **Case ID:** identifies the record that we follow through the system
- **Activity:** describes what happens to the record
- **Timestamp:** describes at which point in time something happened to the record

That means each Event Log is **specific to a certain record type**, e.g. there can be one Event Log for everything related to a Purchase Order Items and another Event Log for Sales Order Items.

The Event Log - or rather the record type we chose as Case ID - **defines the perspective and section of data** for the visualization of the as-is process. A Data Model containing only a single Event Log limits the analysis to this one perspective. With such limitations, we can only analyze one process at a time with a certain focus.

For record types in purchase, sales and production the following limitations can be observed when using single-perspective Data Models:

Purchase Orders Items	Sales Orders Items	Production Order Items	Current Limitation
<ul style="list-style-type: none"> + visualization of all Purchase Requisitions and Invoices related to a Purchase Order - no visibility in "stuck" purchase requisitions - no analysis of invoices w/o PO 	<ul style="list-style-type: none"> + sales order related purchase and production orders can be included - no visibility in buy-/make-to-stock - no connection between a customer's placed orders and claims 	<ul style="list-style-type: none"> + all production orders - no insight in placed sales orders for advanced make-to-stock planning - limited to no insight in parallel and sequential production flows 	<ul style="list-style-type: none"> - different data models with different focus point - no cross-filtering - hard to identify interdependencies and bottlenecks across processes and business areas

Figure 1: Single-perspective Data Models allow for in-depth analysis of single processes but lack inter-process insights

Opportunities with Multi-Event Log Process Mining

Modern organizations of course run a variety of different processes in order to achieve their desired business outcomes. Those processes do not stand alone but interfere with each other. Possible inefficiencies and their root causes are often correlated among different processes. As an example, an inefficiency in the order process (O2C) may be partially or fully caused by an inefficiency happening in the purchasing process (P2P) inside the same organization.

So, while analyzing one process helps to find inefficiencies within that particular process, it does not provide insights into how inefficiencies correlate with events in related processes. Analyzing multiple process perspectives at a time bears the opportunity to capture such dependencies and remove operational friction more holistically.

Multi-Event Log in a nutshell

The Multi-Event Log technology provides the capability to analyze end-to-end business processes and their interdependencies within one analysis. It allows to connect directly linked processes and to bring indirectly linked processes into context.

The Multi-Event Log technology allows you to include multiple Event Logs with their respective case tables in one semantic Data Model. All Event Logs can be linked to each other as well as to master data via joins/foreign key relationships. A holistic analysis can then access all this data on top.

This capability adds value for various use cases. In general there are four high level use cases where Multi-Event Log can be used to simplify or extend the analytical capabilities:

- Use Case 1: Put independent processes into context (filter across processes)
- Use Case 2: Analyze parallel processes by linking multiple hierarchical Event Logs
- Use Case 3: Simplified scripts (no-join transformations) by merging Event Logs
- Use Case 4: Visualize end-to-end processes by linking multiple Event Logs*

By using existing components and process comprehensive filtering you can then analyze how KPIs behave across multiple processes. This enables you to identify inefficiencies and their sources that can be spread over several processes.

* This functionality is only available on the Execution Management System Platform (Cloud deployment) with the new visualisation tech-stack of "Views".

Set up of a Multi-Event Log Data Model

Data Transformation

Writing transformation scripts with Multi-Event Log enabled data models simplifies the transformations as well as can speed up the data transformation (executing the script) significantly.

Instead of having one case key and mapping all activities to this perspective, Multi-Event Log allows to keep different perspectives (that means case definitions) for different processes. For each process, users can simply create a separate activity table with the corresponding case key of that process. This takes away the burden to create complicated joins between source tables for each activity definition. Instead, the link between processes is shifted from the transformation to the Data Model. Their relationship is simply modelled via join key relationships in the visual Data Model Editor.

Let's look at an example, the definition of the 'Clear Invoice' activity in the SAP Purchase-To-Pay process. Usually, this activity is defined in the Purchase-To-Pay process with the following SQL statement:

```
SELECT DISTINCT
  E._CASE_KEY AS "_CASE_KEY"
  ,E.MANDT AS "MANDT"
  ,E.EBELN AS "EBELN"
  ,E.EBELP AS "EBELP"
  , 'Gleiche Rechnung aus' AS "ACTIVITY_DE"
  , 'Clear Invoice' AS "ACTIVITY_EN"
  ,CASE
    WHEN CAST(BKPF_Z."CPUDT" AS DATE) = CAST(B."AUGDT" AS DATE)
    THEN CAST(BKPF_Z."CPUDT" AS DATE) + CAST(BKPF_Z."CPUTM" as TIME)
    ELSE CAST(B."AUGDT" AS DATE) + CAST('23:59:59' as TIME)
  END AS "EVENTTIME"
  ,2500 AS "_SORTING"
  ,BKPF_Z.USNAM AS "USER_NAME"
  ,USR02.USTYP AS "USER_TYPE"
```

```

, BKPF_Z.TCODE AS "TRANSACTION_CODE"
FROM
RSEG AS RSEG
INNER JOIN TMP_P2P_EKKO_EKPO AS E ON 1=1
    AND RSEG.MANDT = E.MANDT
    AND RSEG.EBELN = E.EBELN
    AND RSEG.EBELP = E.EBELP
INNER JOIN TMP_P2P_BKPF_BSEG AS B ON 1=1
    AND B.MANDT = RSEG.MANDT
    AND SUBSTRING(B.AWKEY, 1, 14) = RSEG.BELNR || CAST(RSEG.GJAHR AS
VARCHAR(4))
LEFT JOIN BKPF AS BKPF_Z ON 1=1
    AND B.MANDT = BKPF_Z.MANDT
    AND B.BUKRS = BKPF_Z.BUKRS
    AND B.AUGBL = BKPF_Z.BELNR
    AND B.AUGGJ = BKPF_Z.GJAHR
LEFT JOIN USR02 AS USR02 ON 1=1
    AND BKPF_Z.MANDT = USR02.MANDT
    AND BKPF_Z.USNAM = USR02.BNAME
WHERE
B.AUGBL IS NOT NULL AND
BKPF_Z.CPUDT IS NOT NULL;

```

The details like the 'User Name' for the 'Clear Invoice' activity are not stored in the Purchase Order table (EKPO). They are stored in the Accounting Document Header table (BKPF). To add this information, it is necessary to create this lengthy join starting from EKKO/EKPO to RSEG to BKPF/BSEG to the clearing document in BKPF (declared as BKPF_Z).

With the Multi-Event Log approach, users can save lines of code and associated headache. The clearing activity can be written together with all other invoice related activities into a separate activity table with the case key of the corresponding invoice document. This way, the table join is much shorter. The link to a purchase document is not necessary at this point. The SQL statement can be reduced to the following expression:

```

SELECT DISTINCT
  B."_CASE_KEY" AS "_CASE_KEY"
  , 'Clear Invoice' AS "ACTIVITY_EN"
  , CASE
      WHEN CAST(BKPF_Z."CPUDT" AS DATE) = CAST(B."AUGDT" AS DATE)
      THEN CAST(BKPF_Z."CPUDT" AS DATE) + CAST(BKPF_Z."CPUTM" as TIME)
      ELSE CAST(B."AUGDT" AS DATE) + CAST('23:59:59' as TIME)
    END AS "EVENTTIME"
  , BKPF_Z."USNAM" AS "USER_NAME"
  , USR02."USTYP" AS "USER_TYPE"
  , 900 AS "_SORTING"
  , B."MANDT" AS "MANDT"
  , B."BUKRS" AS "BUKRS"
  , B."BELNR" AS "BELNR"
  , B."GJAHR" AS "GJAHR"
  , B."BUZEI" AS "BUZEI"
  , BKPF_Z."TCODE" AS "TRANSACTION_CODE"
FROM
  "TMP_AP_BKPF_BSEG" AS B
  LEFT JOIN "BKPF" AS BKPF_Z ON 1=1
    AND B."MANDT" = BKPF_Z."MANDT"
    AND B."BUKRS" = BKPF_Z."BUKRS"
    AND B."AUGBL" = BKPF_Z."BELNR"
    AND B."AUGGJ" = BKPF_Z."GJAHR"
  LEFT JOIN "USR02" AS USR02 ON 1=1
    AND BKPF_Z."MANDT" = USR02."MANDT"
    AND BKPF_Z."USNAM" = USR02."BNAME"
WHERE
  B.AUGBL IS NOT NULL AND
  BKPF_Z.CPUDT IS NOT NULL;

```

The example above contains two instead of the previous four table joins. But writing scripts this way, not only simplifies the necessary joins. Using a separate activity table for invoice documents allows users to additionally include invoices in the Data Model that do not have a link to a purchase document. For the SAP example, invoices in the Data Model are not restricted

to MM related invoices (link to Purchase Order). FI invoices are also part of the Data Model which would otherwise only be visible in the Accounts Payable process.

Data Model

The setup of a Multi-Event Log Data Model is similar to the setup of a Data Model with one Event Log. First, all tables of a Data Model need to be added. Afterwards, the foreign key relationships between tables need to be defined. Now, every table that serves as an Event Log can be configured as an Activity table by choosing the role “Activities table” shown in the following screenshot.

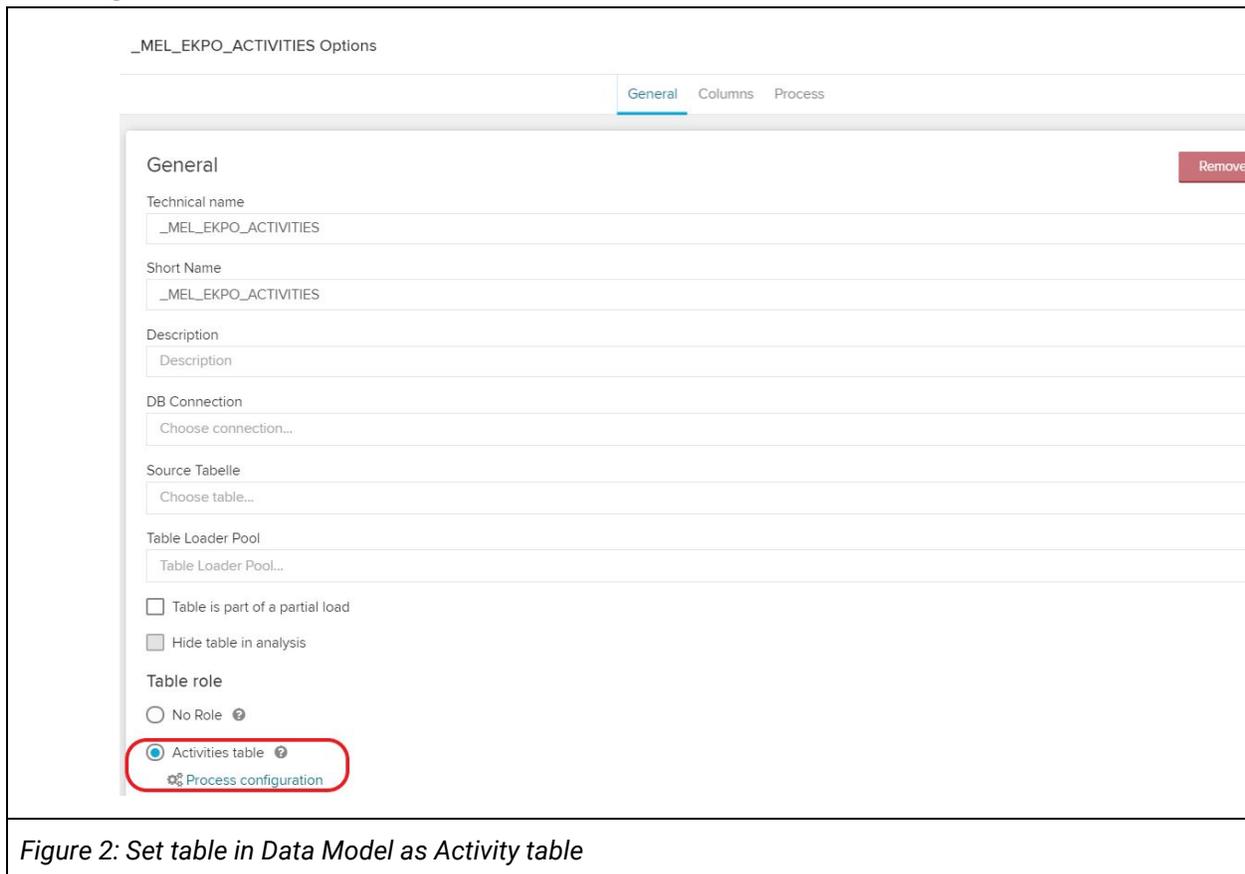


Figure 2: Set table in Data Model as Activity table

After choosing the option, the process configuration has to be performed and Case ID, Activity Name and Timestamp columns can be defined as usual. **Additionally, the corresponding case table has to be set in this configuration step (see figure 3).** One Activity Table can be defined as the default Activity Table.

Afterwards, the configured activity table is marked with an **A** in the Data Model. The default Activity Table is marked as **DA**. Case tables are indicated with a **C**.

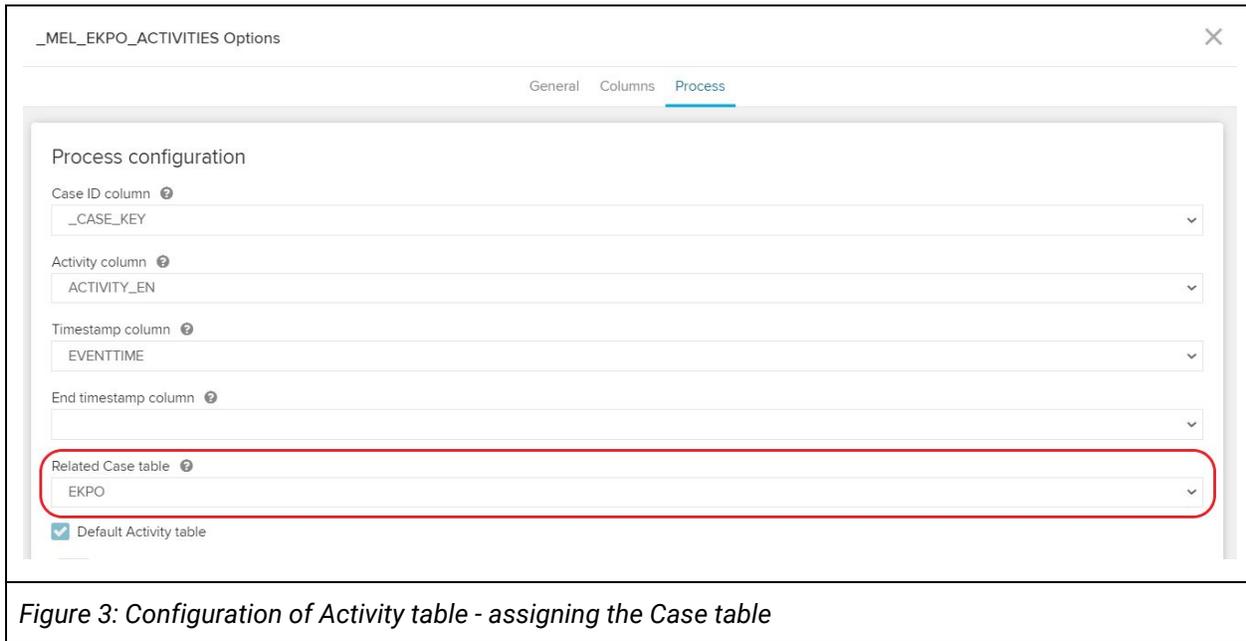


Figure 3: Configuration of Activity table - assigning the Case table

Set up of a Multi-Event Log Analysis

Discover multiple processes

Once the Data Model is loaded, the analysis can be set up to discover all processes (Event Logs) in context with each other (Use case 1).

Configure Process Visualization in Process Explorer and Variant Explorer

Within the analysis, all process related components like the Process Explorer, Variant Explorer or Case Explorer refer to the Activity table marked as the Default Activity table in the Data Model (see previous chapter). However, the Variant Explorer as well as the Process Explorer allow to dynamically choose the Event Log that is visualized by selecting in the settings a “Custom dimension”. This dimension refers by default to the specified activity column of the default activity table. By ticking the “Custom dimension” option it can be changed in the PQL Editor to any column of any Activity table in the Data Model.

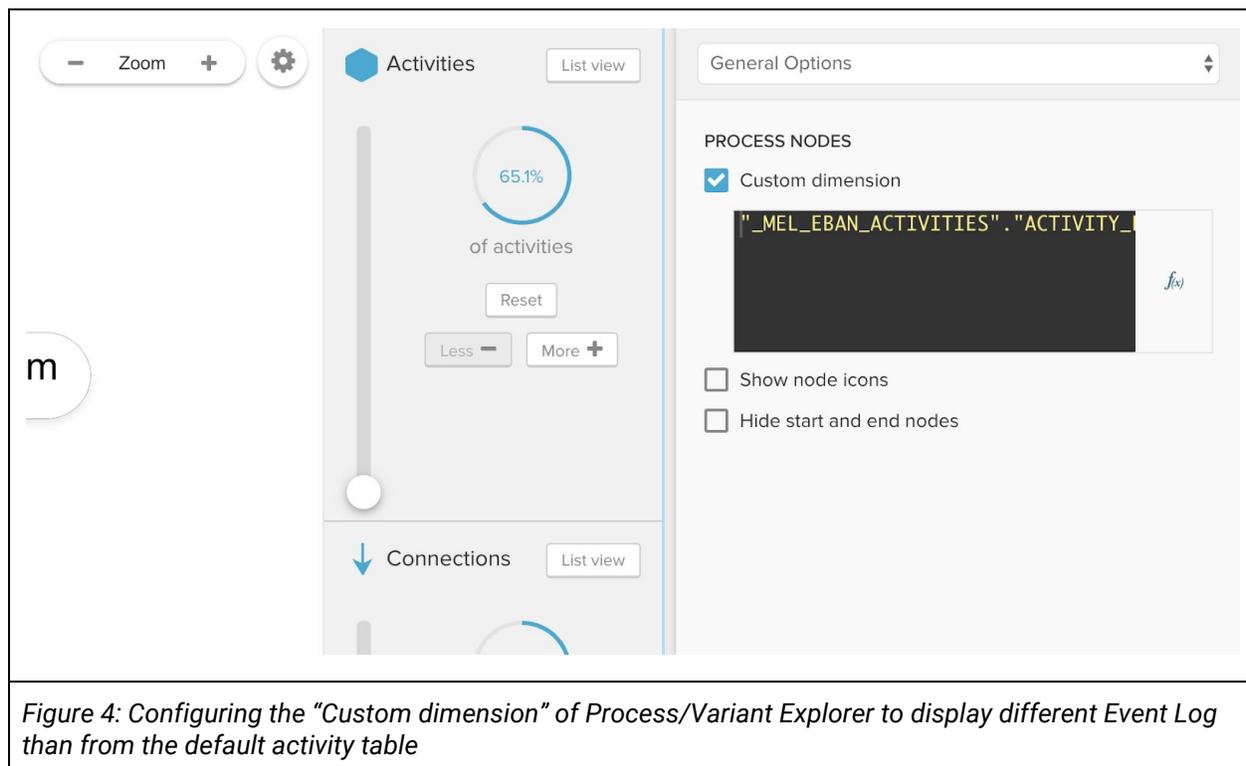


Figure 4: Configuring the “Custom dimension” of Process/Variant Explorer to display different Event Log than from the default activity table

In the below example, the two Process Explorers have custom dimensions, one referencing the activity column of the Order-To-Cash activity table and one referencing the activity column of the Purchase-To-Pay activity table.

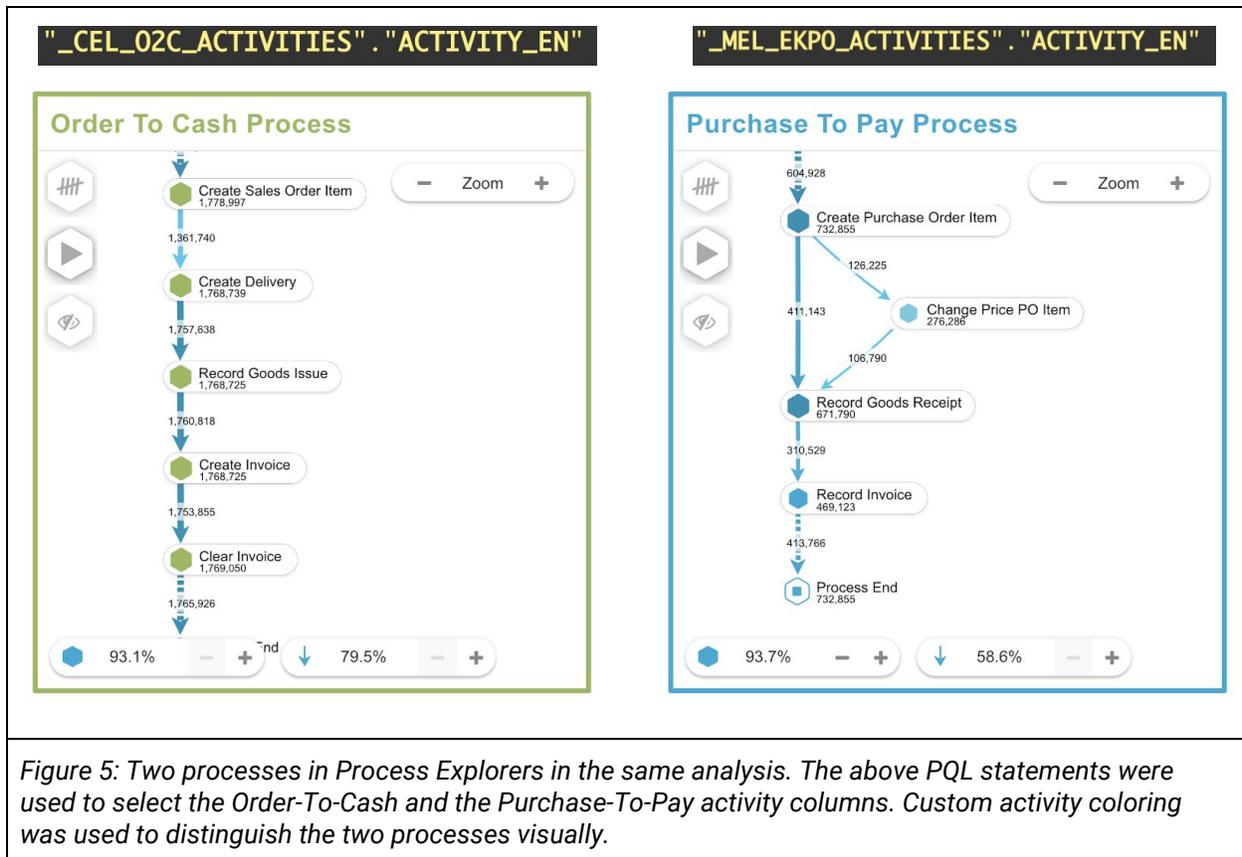


Figure 5: Two processes in Process Explorers in the same analysis. The above PQL statements were used to select the Order-To-Cash and the Purchase-To-Pay activity columns. Custom activity coloring was used to distinguish the two processes visually.

Furthermore, users can now even combine two Event Logs into one Event Log with the PQL Function MERGE_EVENTLOG. Users that split for example the Purchase Requisition activities from the Purchase Order activities into two Event Logs, to save time writing lengthy transformations per activity, can now combine the two activities into one Event Log by merging them together. For further details on this function, please visit the PQL Documentation on <https://help.celonis.de/>

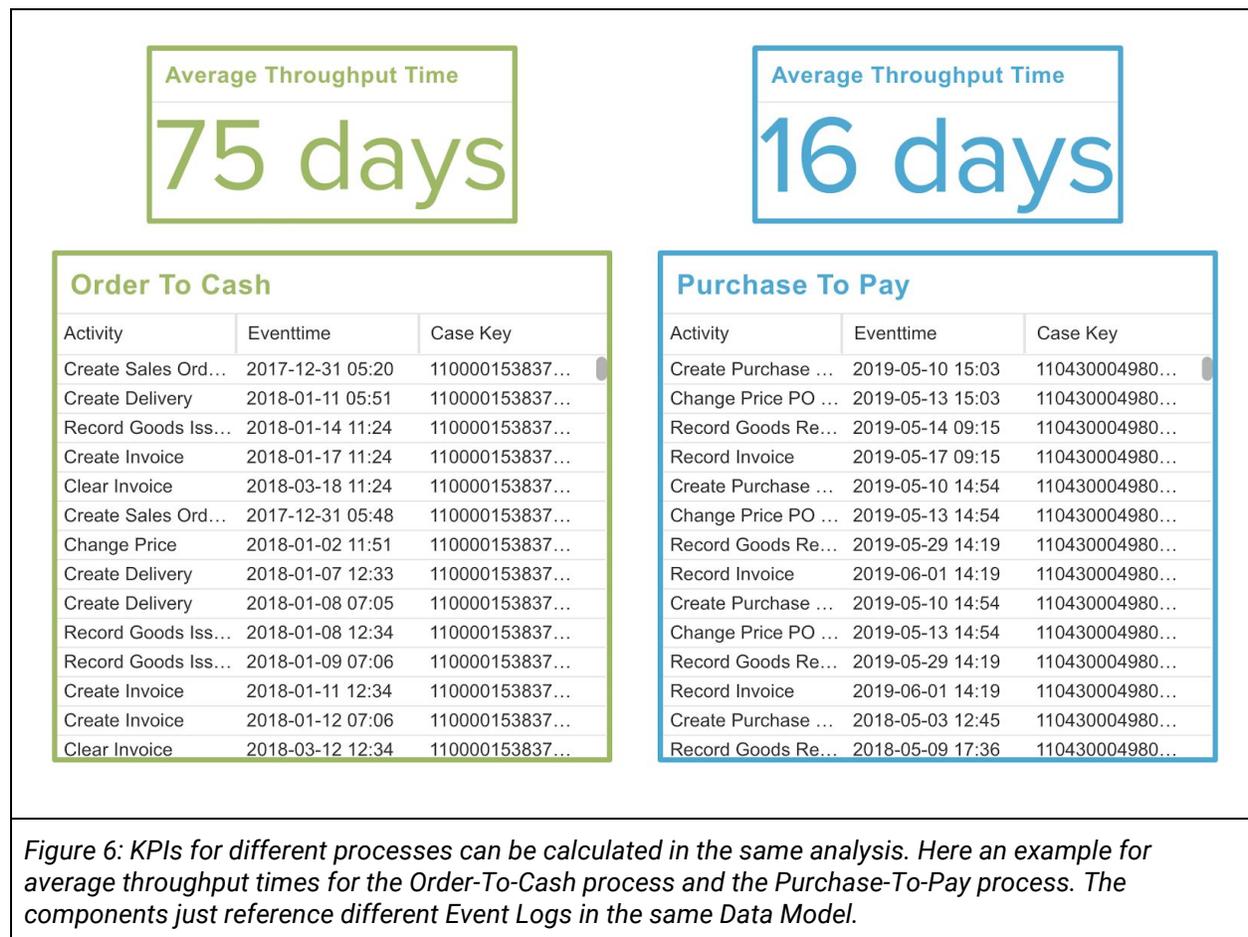
Create KPI Calculations across processes

Of course it is also possible to

- create KPIs for different processes within one analysis,
- filter across multiple processes, and
- create KPIs across multiple processes.

Metrics from different Event Logs in one Analysis

Here is an example of metrics of two processes in the same analysis:



Filter across multiple processes

When the cases of different processes are connected, users can filter and discover dependencies across processes.

The example below shows that by filtering on one Purchase Order Item also the Order-To-Cash Process is filtered and the related Sales Order Item to this Purchase Order Item is shown.

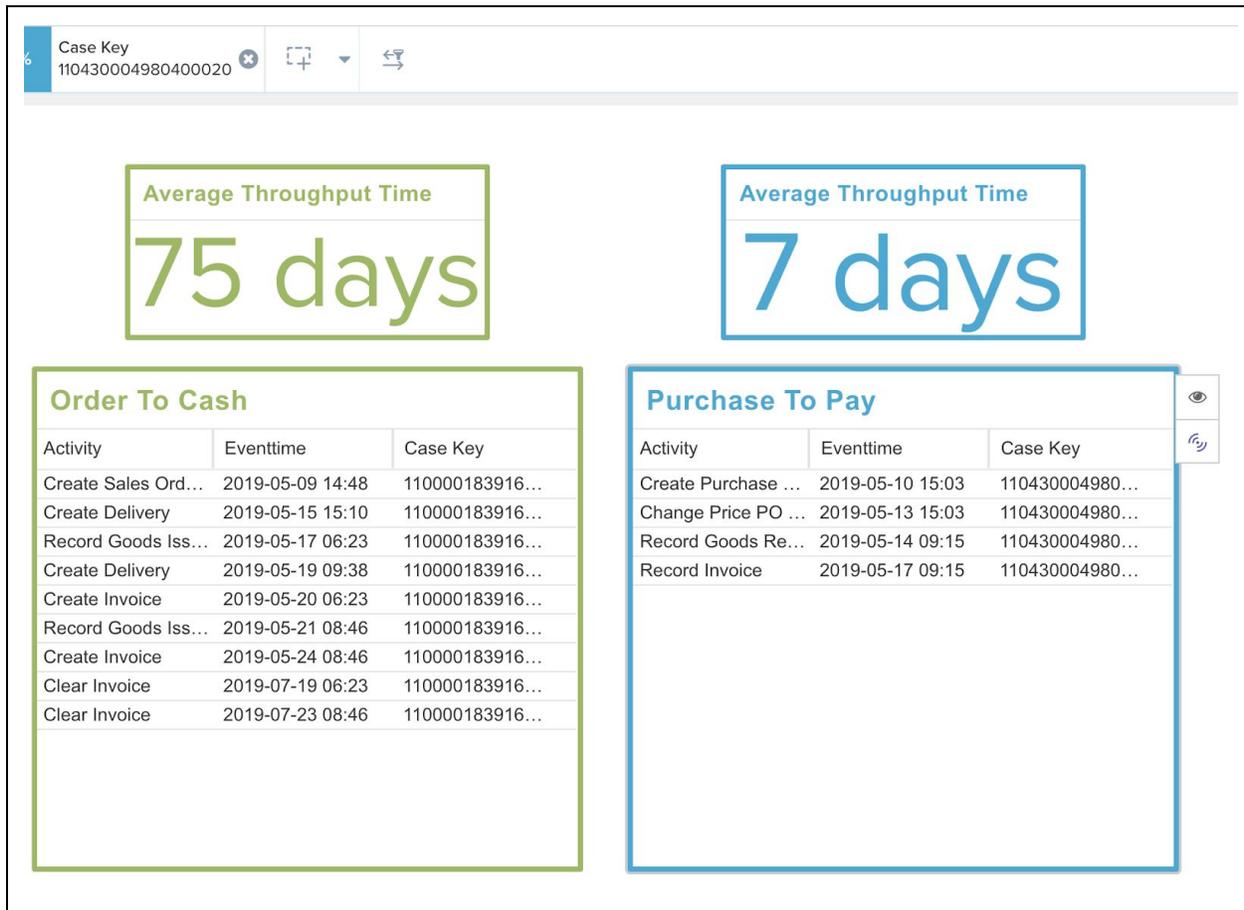


Figure 7: For Multi-Event Log Data Models the filter logic is the same as before.

Metrics across different Event Logs

There are no limitations to calculate cross-process statistics. Typical questions could be:

- How do KPIs between different processes correlate? How often for example are complaints in the service process filed for parts produced in the manufacturing process?
- How do process KPIs like Throughput Time correlate across processes?

- How much time passes from one process to another?

As we have seen in the filtering example, the Celonis PQL behaviour is the same for Multi-Event Log as before. To compare the throughput times, one could simply calculate the throughput times per process and then choose from the over 150 PQL functions to run statistical analysis on both data sets.

Discover parallel processes

One main advantage of the Multi-Event Log technology is the ability to model and work with parallel processes (Use case 2). Parallel processes can always be looked at in a hierarchical way. An example for a hierarchical process is this car manufacturing process.

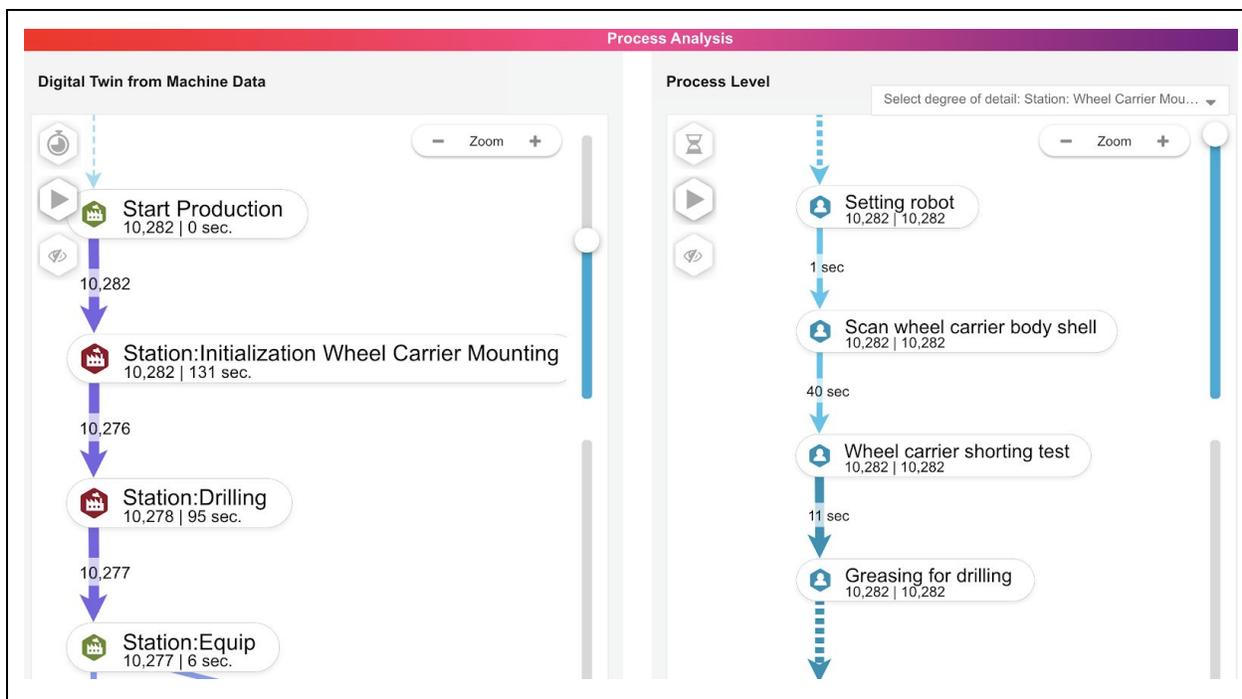


Figure 8: High-level process on the left. Granular process for each station on the right. The activities on the right are all taking place inside the activity "Station: Initialization Wheel Carrier Mounting" of the high-level process.

The top hierarchy of the process is modeled on station level. Each station is represented by one activity. Inside each station are multiple activities happening. This more granular process is in the Data Model stored in a separate Activity table and represented on the right in a separate Process Explorer.

This way of modelling can be exploited for data exploration of parallel processes. If there are parallel streams on the granular process level, they can now be visualized natively in this granular Process Explorer. As we use the granular Event Log for this Process Explorer, multiple cases can occur in parallel for each high level case. An example of this is shown below for a review process that includes two reviewers:

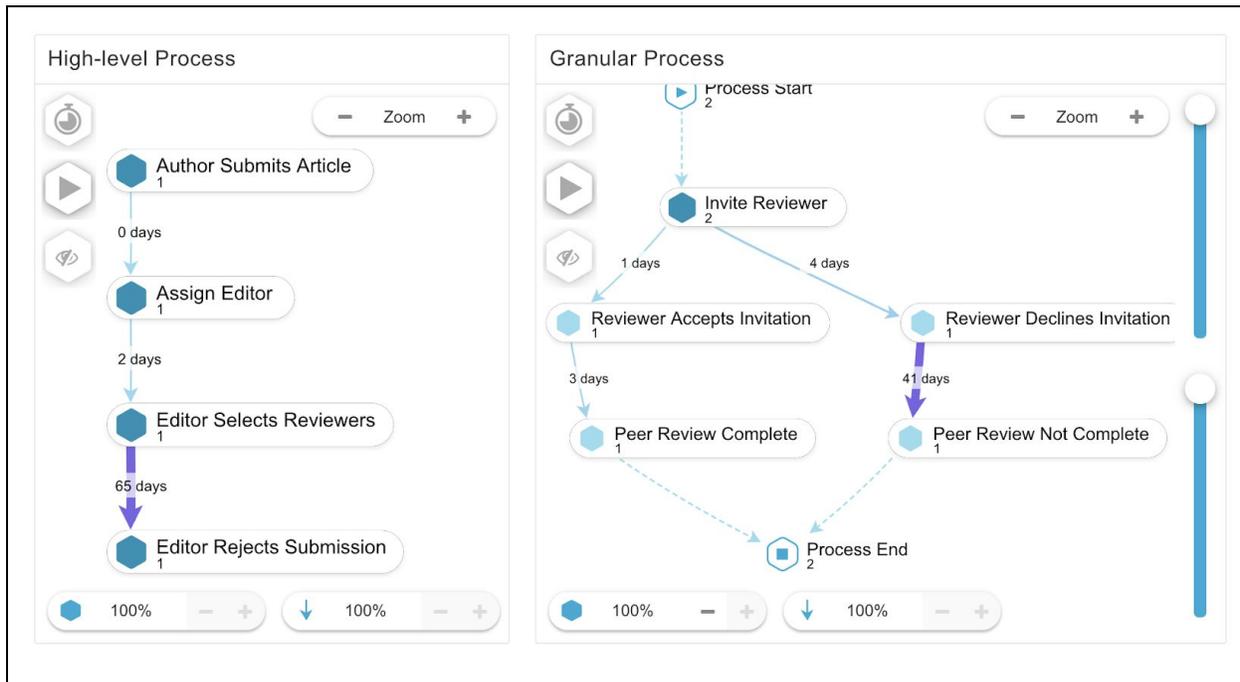


Figure 9: The high-level process on the left has one case per reviewed paper. The granular reviewing process on the right has one case per reviewer. Note that for one high-level case, multiple granular cases can occur in parallel. The Multi-Event Log technology allows to visualize and calculate KPIs on parallel processes.

To set up such a hierarchical Data Model that allows to analyze parallel processes, the Event Logs of different granularity can be linked through the Case tables. The number of levels of granularity is not limited. It would for example be possible to have four or five levels of granularity.

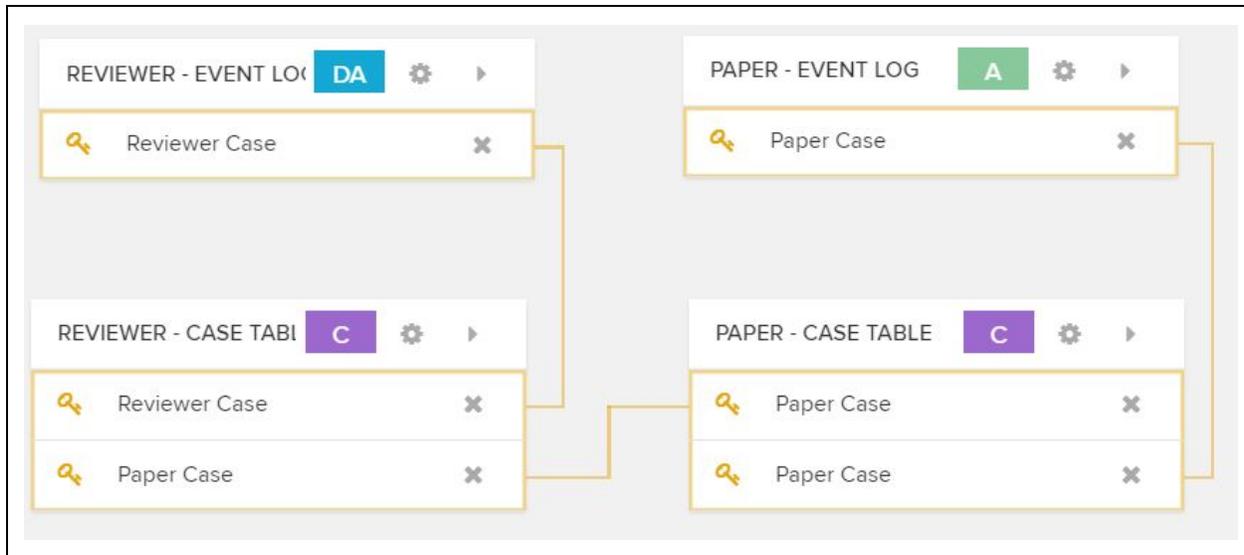


Figure 10: Typical set up to analyze parallel processes inside the granular process, here the Reviewer process. Each case of the granular process (Reviewer Case) is linked to a case of the high-level process (Paper Case).

Limitations of the current Analysis regarding Multi-Event Log usage

Existing components and visualizations in the Analysis UI can be used for Multi-Event Log analyses to a great extent already. Some components are more applicable to single Event Log use cases and always access the default Activity table/Case table, those are: Throughput Time Search, Activity Explorer, Case Explorer, Selection Views (except Attribute Selection), Conformance Checker.

Some best practices to leverage the existing Analysis capabilities for MEL analyses:

- **Throughput Time Search:** can be composed with variables, button drilldowns, single KPI and histogram to calculate throughput times for a specific or across Event Logs
- **Process Explorer:**
 - Use Case 1: Put independent processes into context (filter across processes) - Use one process explorer for each Event Log and define "Custom dimension" accordingly
 - Use Case 2: Use MERGE_EVENTLOG functionality in "Custom dimension"
- **Case Explorer:** can be composed for each Case & Activity Table on a sheet with two OLAP tables, one holding the case information and one holding the activity information