Object-Centric Event Log 2.0

OCEL 2.0

Alessandro Berti, Istvan Koren, Jan Niklas Adams, Gyunam Park, Benedikt Knopp, Nina Graves, Majid Rafiei, Lukas Liß, Leah Tacke Genannt Unterberg, Yisong Zhang, Christopher Schwanen, Marco Pegoraro, and Wil van der Aalst

© Wil van der Aalst (use only with permission & acknowledgements)

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

cute
cows

Why OCPM?

Avoid repeatedly going back to your source systems

(a system-agnostic single source of truth)

See and understand the interactions between different object types

(problems live at the intersections of processes and organizational entities)

Avoid distortions due to the single-case assumption

(circumventing convergence and divergence problems)

© Wil van der Aalst (use only with permission & acknowledgements)

Why OCEL 2.0?

Concrete!

Balancing simplicity and expressiveness

Comes with a formal definition

Easy to understand meta-model

Three exchange formats: XML, JSON, Relational

Software libraries to support developers

Several publically available datasets

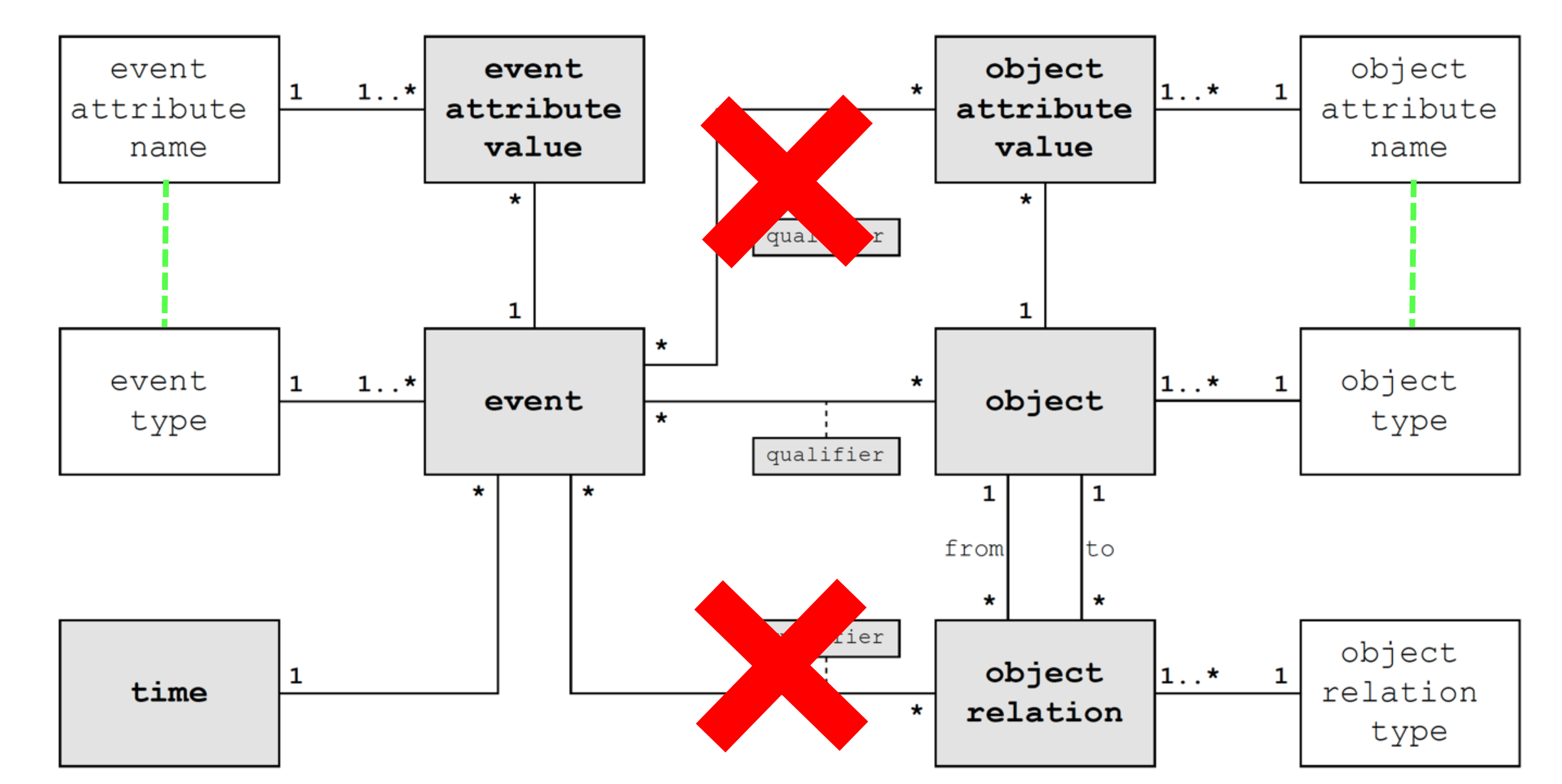
Supported by multiple process mining tools

A horse, not a camel ☺

© Wil van der Aalst (use only with permission & acknowledgements)

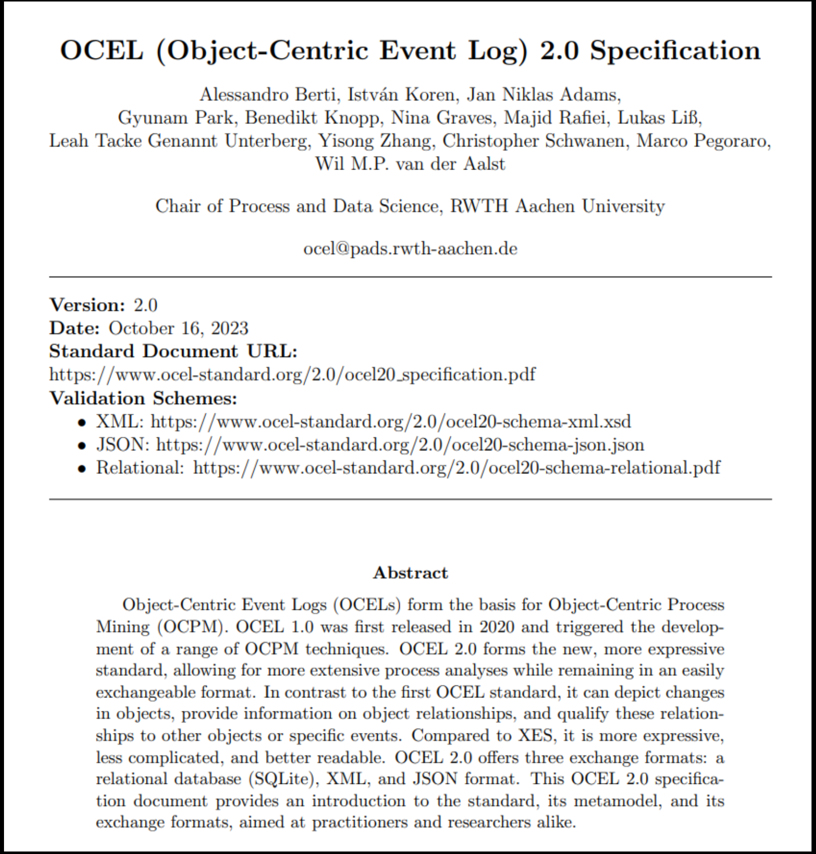
Main differences with the extended OCED meta-model

Only one path to navigate from events to objects instead of four



© Wil van der Aalst (use only with permission & acknowledgements)

Object-Centric Event Log (OCEL) 2.0



OCEL (Object-Centric Event Log) 2.0 Specification

Alessandro Berti, István Koren, Jan Niklas Adams,
Gyunam Park, Benedikt Knopp, Nina Graves, Majid Rafiei, Lukas Liß,
Leah Tacke Genannt Unterberg, Yisong Zhang, Christopher Schwanen, Marco Pegoraro,
Wil M.P. van der Aalst

Chair of Process and Data Science, RWTH Aachen University

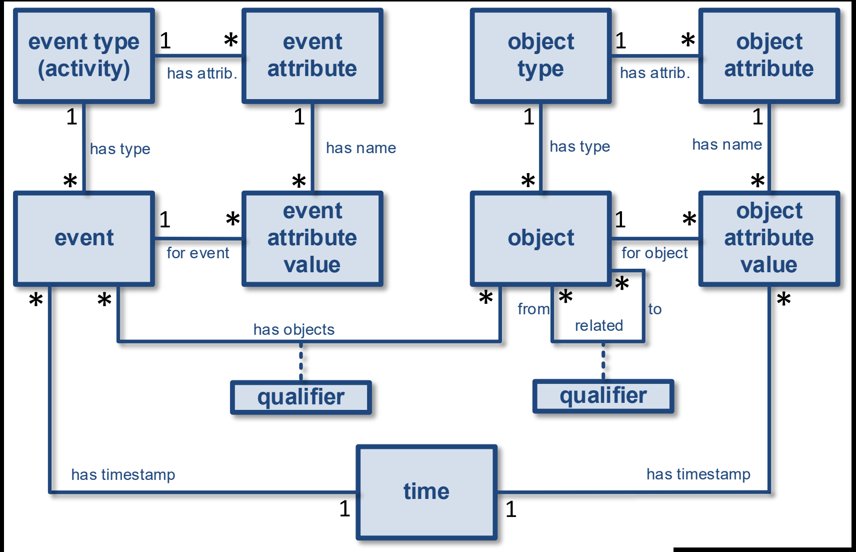
ocel@pads.rwth-aachen.de

Version: 2.0
Date: October 16, 2023
Standard Document URL:
https://www.ocel-standard.org/2.0/ocel20_specification.pdf
Validation Schemes:

- XML: https://www.ocel-standard.org/2.0/ocel20-schema-xml.xsd
- JSON: https://www.ocel-standard.org/2.0/ocel20-schema-json.json
- Relational: https://www.ocel-standard.org/2.0/ocel20-schema-relational.pdf

Abstract

Object-Centric Event Logs (OCELs) form the basis for Object-Centric Process Mining (OCPM). OCEL 1.0 was first released in 2020 and triggered the development of a range of OCPM techniques. OCEL 2.0 forms the new, more expressive standard, allowing for more extensive process analyses while remaining in an easily exchangeable format. In contrast to the first OCEL standard, it can depict changes in objects, provide information on object relationships, and qualify these relationships to other objects or specific events. Compared to XES, it is more expressive, less complicated, and better readable. OCEL 2.0 offers three exchange formats: a relational database (SQLite), XML, and JSON format. This OCEL 2.0 specification document provides an introduction to the standard, its metamodel, and its exchange formats, aimed at practitioners and researchers alike.



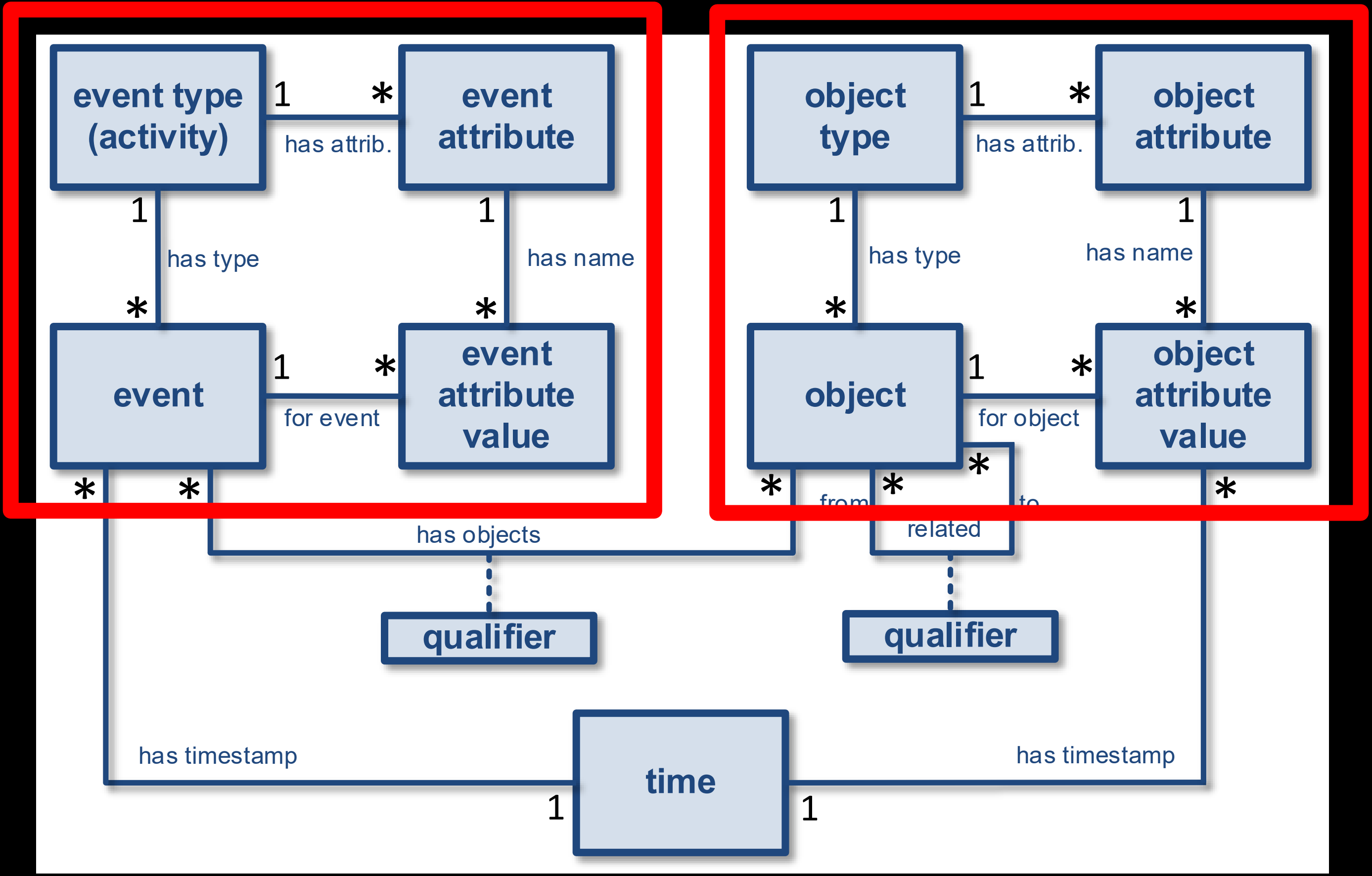
OCEL STANDARD

Based on extensive experiences with OCEL 1.0 standard released in 2020!

https://www.ocel-standard.org/

© Wil van der Aalst (use only with permission & acknowledgements)

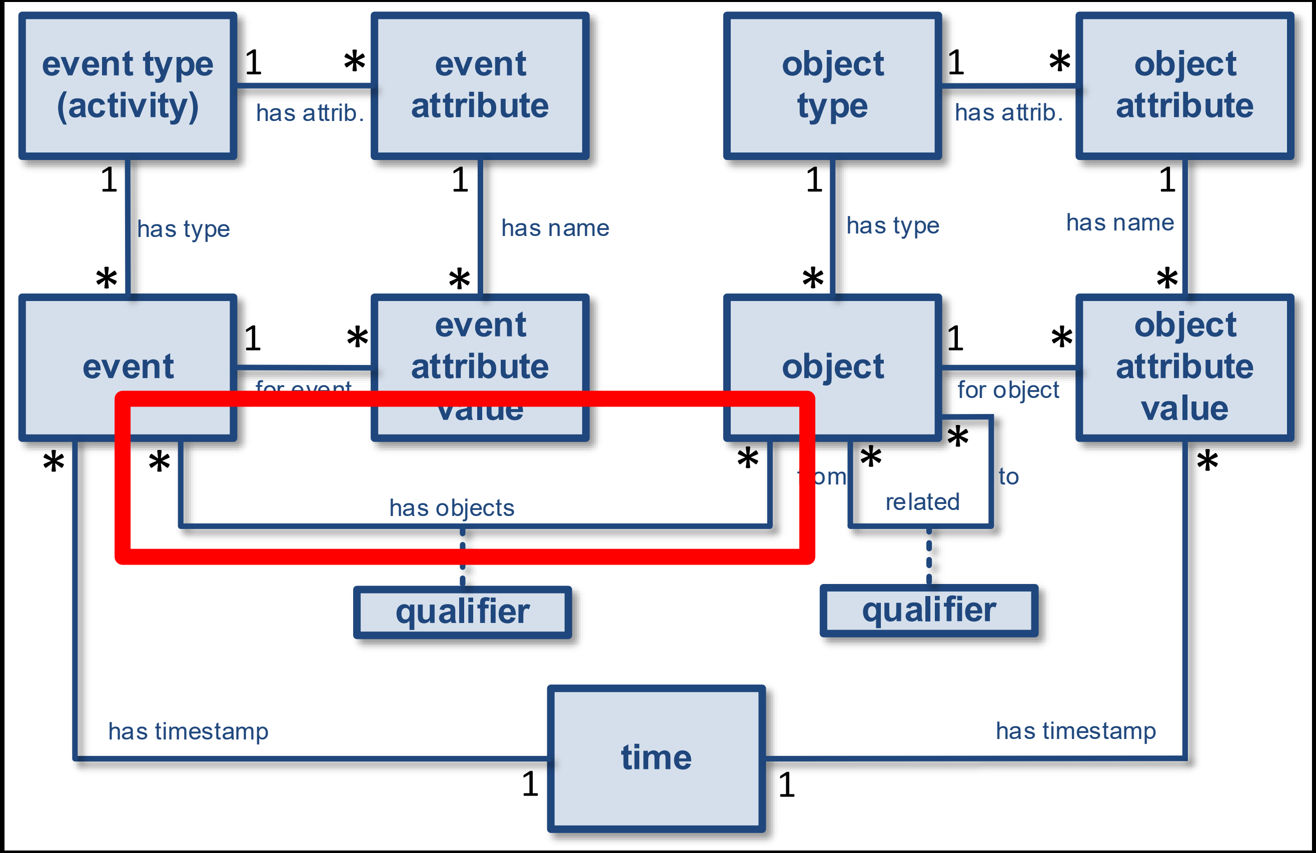
Events and Objects



© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

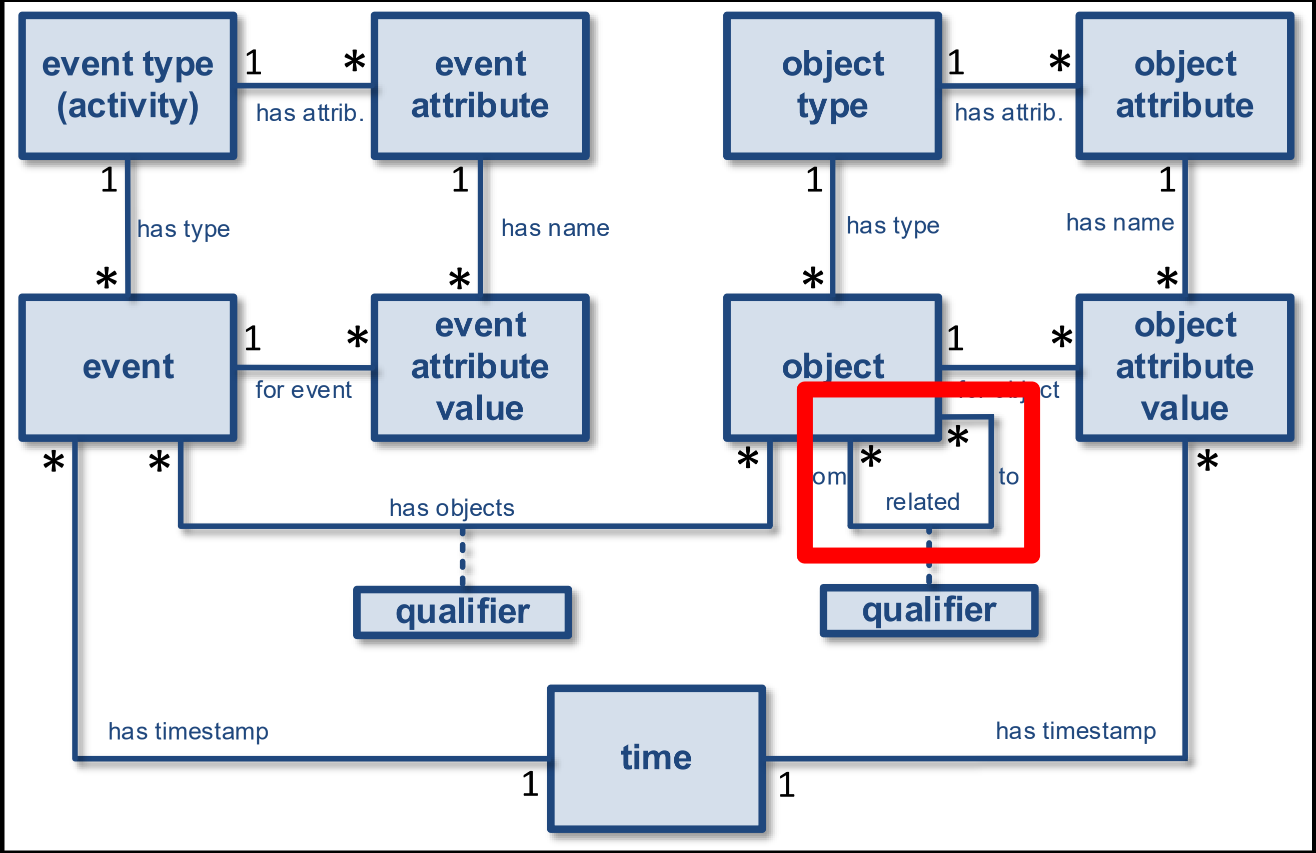
Event-to-Object (E2O) Relations



© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

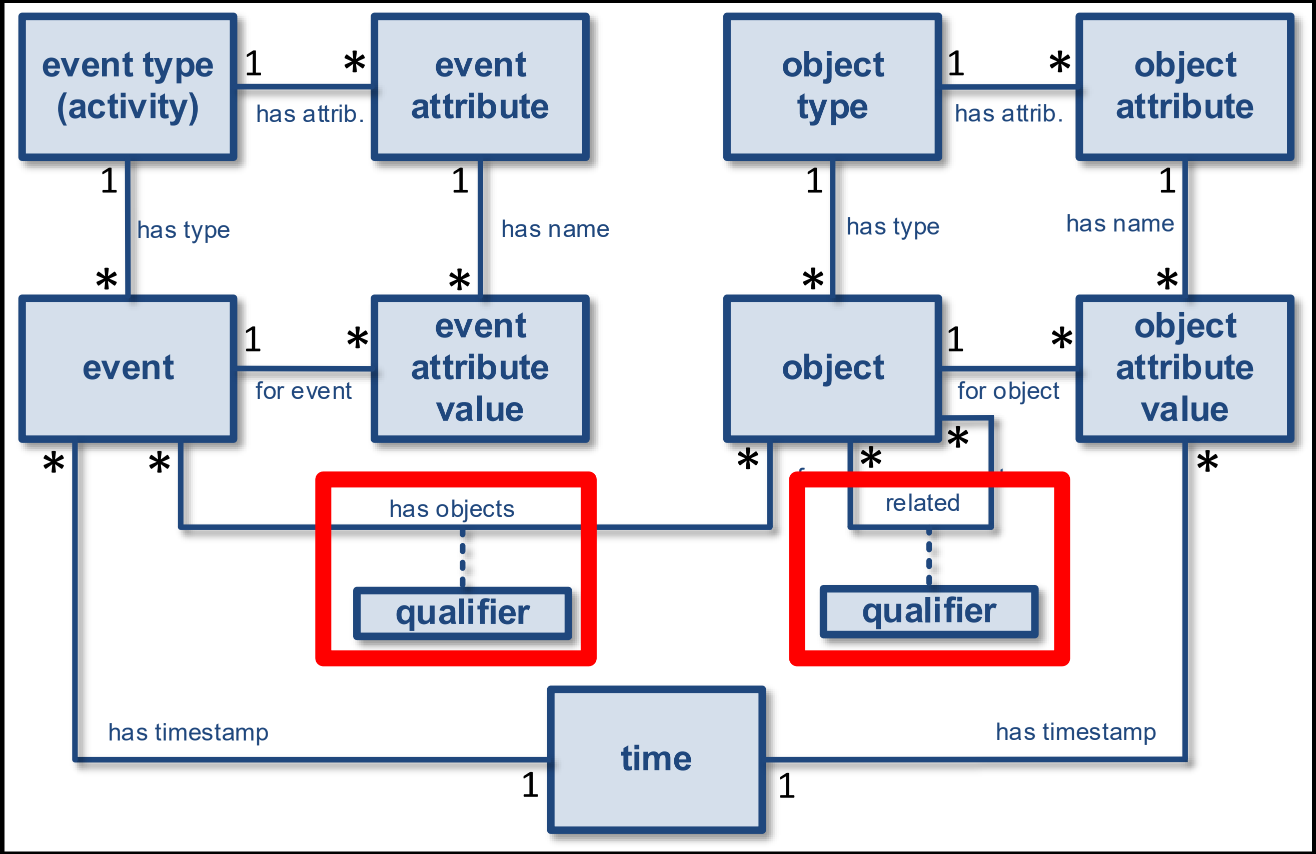
Object-to-Object (O2O) Relations (New!)



© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

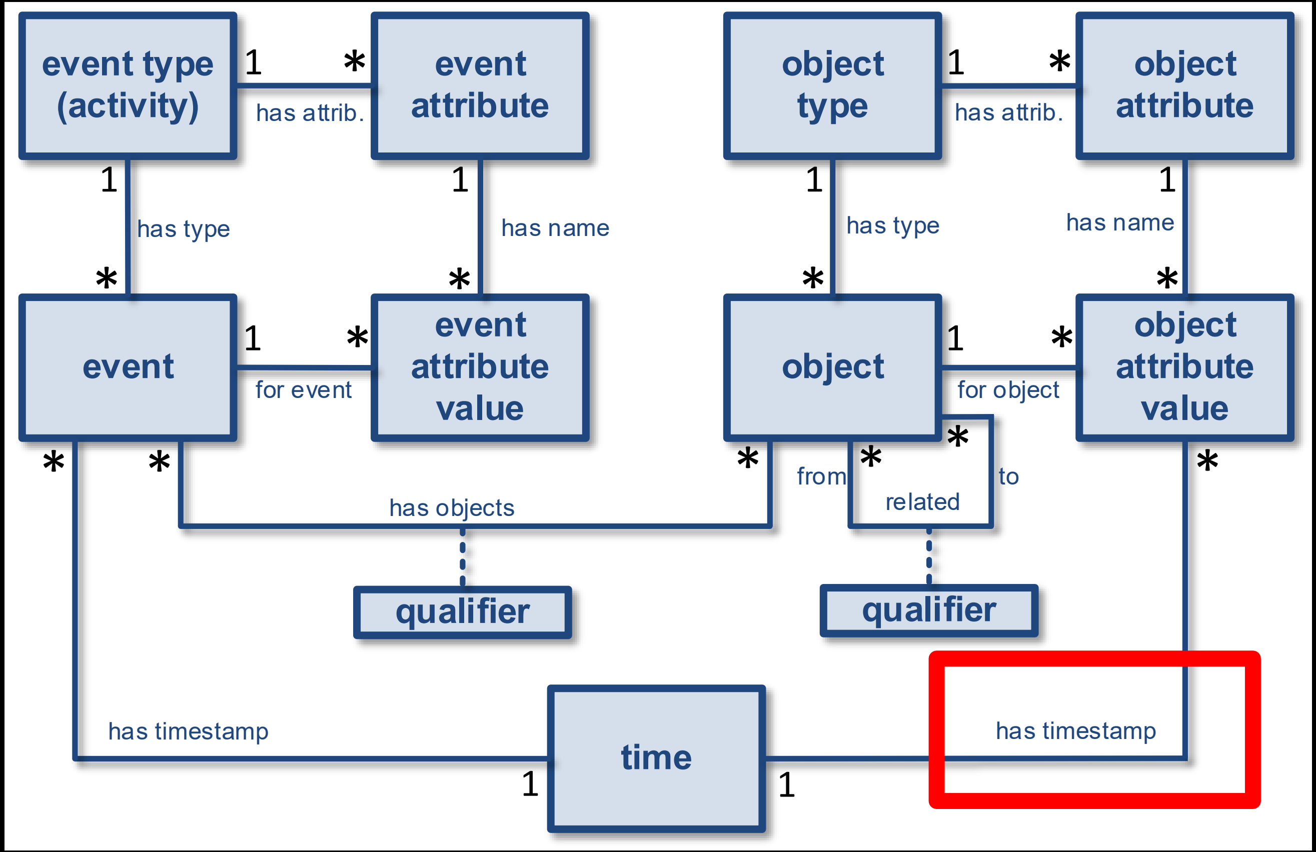
Qualified Relations (New!)



© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

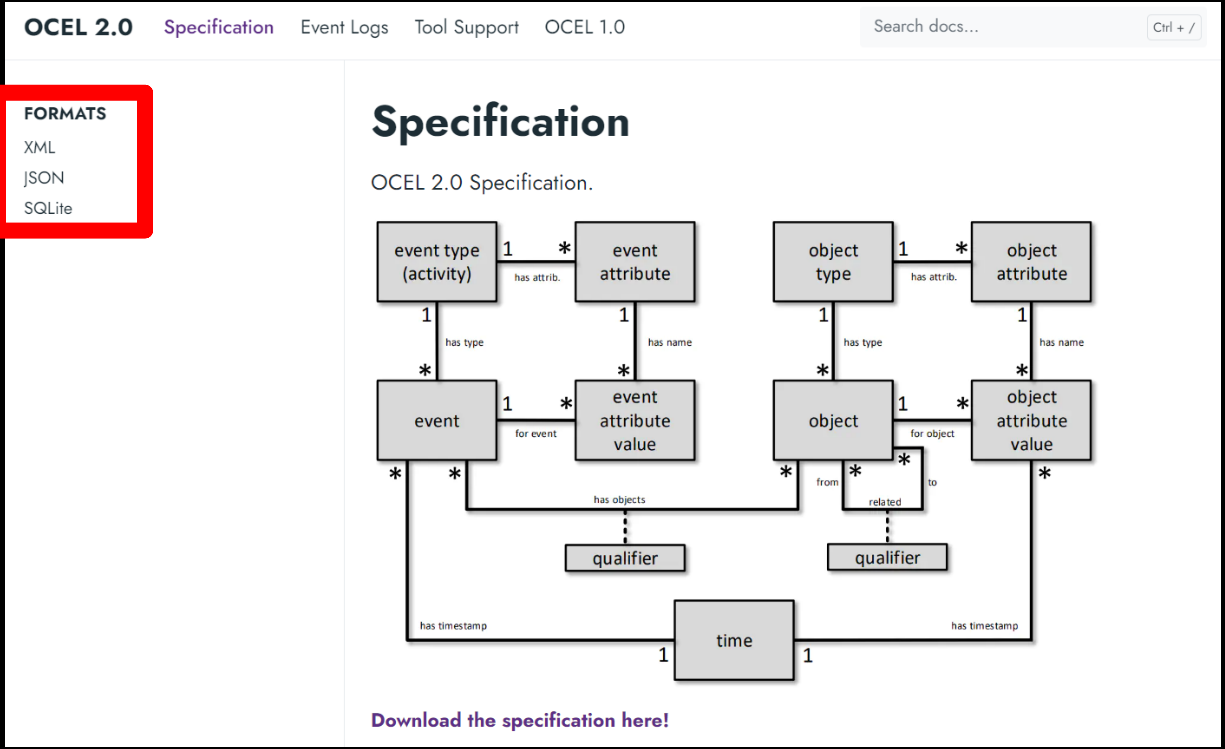
Object attributes can change over time (New!)



© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

Three exchange formats: XML, JSON, Relational (New!)



OCEL 2.0 Specification Event Logs Tool Support OCEL 1.0 Search docs... Ctrl + /

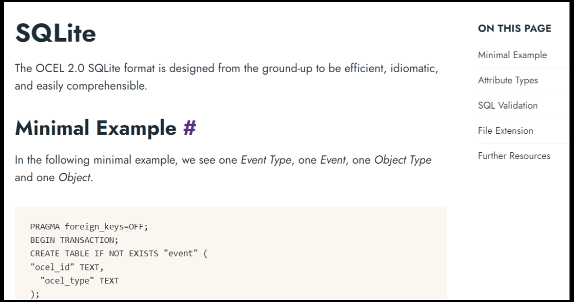
FORMATS

- XML
- JSON
- SQLite

Specification

OCEL 2.0 Specification.

Download the specification here!



SQLite

The OCEL 2.0 SQLite format is designed from the ground-up to be efficient, idiomatic, and easily comprehensible.

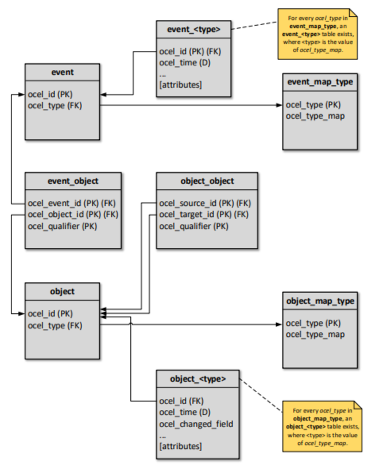
Minimal Example

In the following minimal example, we see one *Event Type*, one *Event*, one *Object Type* and one *Object*.

```
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE IF NOT EXISTS "event" (
"ocel_id" TEXT,
  "ocel_type" TEXT
);
```

ON THIS PAGE

- Minimal Example
- Attribute Types
- SQL Validation
- File Extension
- Further Resources

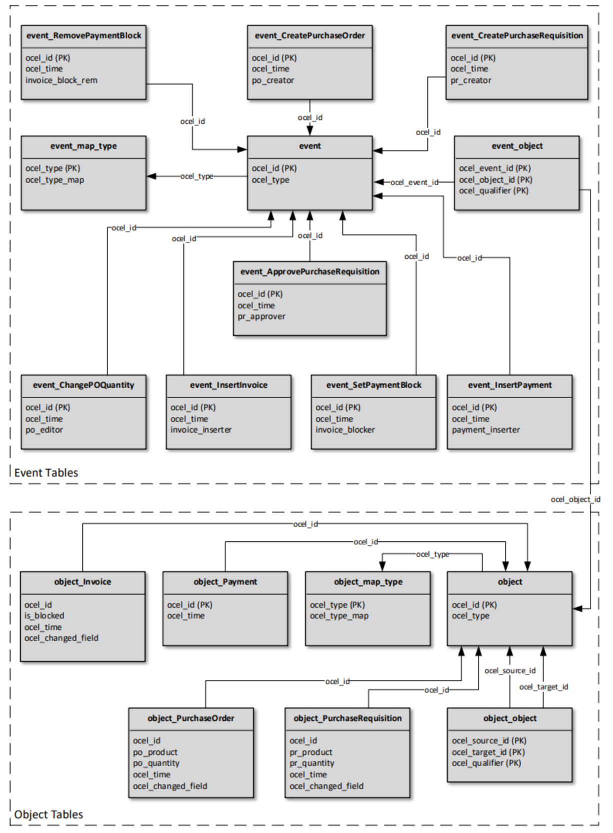


© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

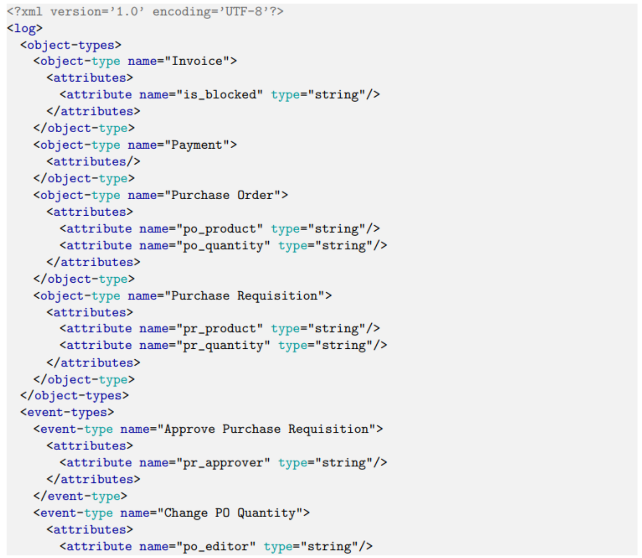
Three exchange formats (New!)

Relational (SQLite)



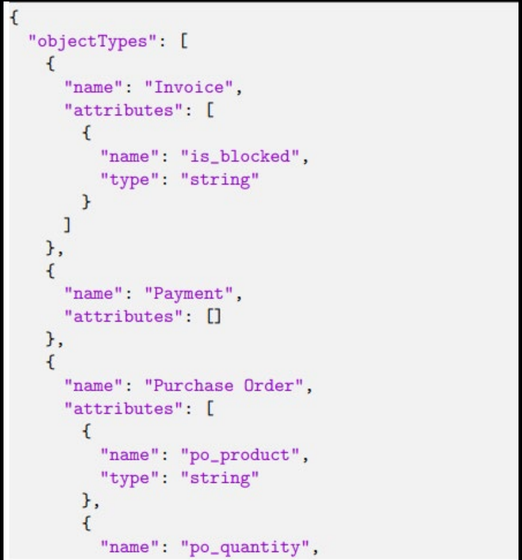
Tools for automated checks are provided

XML



```xml
<?xml version='1.0' encoding='UTF-8'?>
<log>
  <object-types>
    <object-type name="Invoice">
      <attributes>
        <attribute name="is_blocked" type="string"/>
      </attributes>
    </object-type>
    <object-type name="Payment">
      <attributes/>
    </object-type>
    <object-type name="Purchase Order">
      <attributes>
        <attribute name="po_product" type="string"/>
        <attribute name="po_quantity" type="string"/>
      </attributes>
    </object-type>
    <object-type name="Purchase Requisition">
      <attributes>
        <attribute name="pr_product" type="string"/>
        <attribute name="pr_quantity" type="string"/>
      </attributes>
    </object-type>
  </object-types>
  <event-types>
    <event-type name="Approve Purchase Requisition">
      <attributes>
        <attribute name="pr_approver" type="string"/>
      </attributes>
    </event-type>
    <event-type name="Change PO Quantity">
      <attributes>
        <attribute name="po_editor" type="string"/>
```

JSON

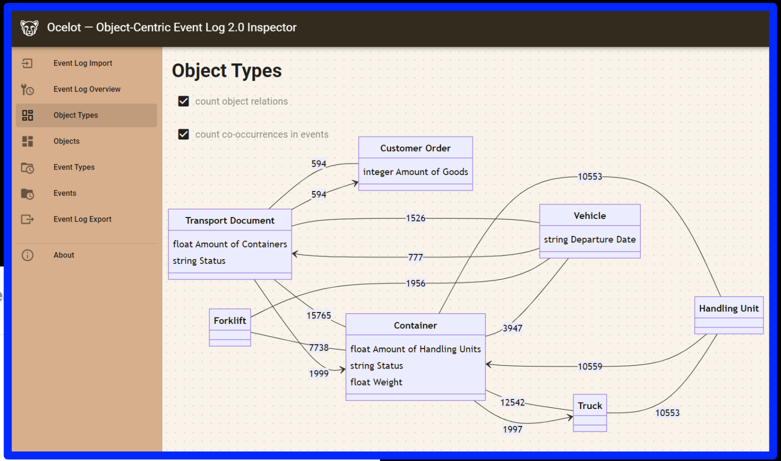


```json
{
  "objectTypes": [
    {
      "name": "Invoice",
      "attributes": [
        {
          "name": "is_blocked",
          "type": "string"
        }
      ]
    },
    {
      "name": "Payment",
      "attributes": []
    },
    {
      "name": "Purchase Order",
      "attributes": [
        {
          "name": "po_product",
          "type": "string"
        },
        {
          "name": "po_quantity",
```

© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

Event logs (New!)

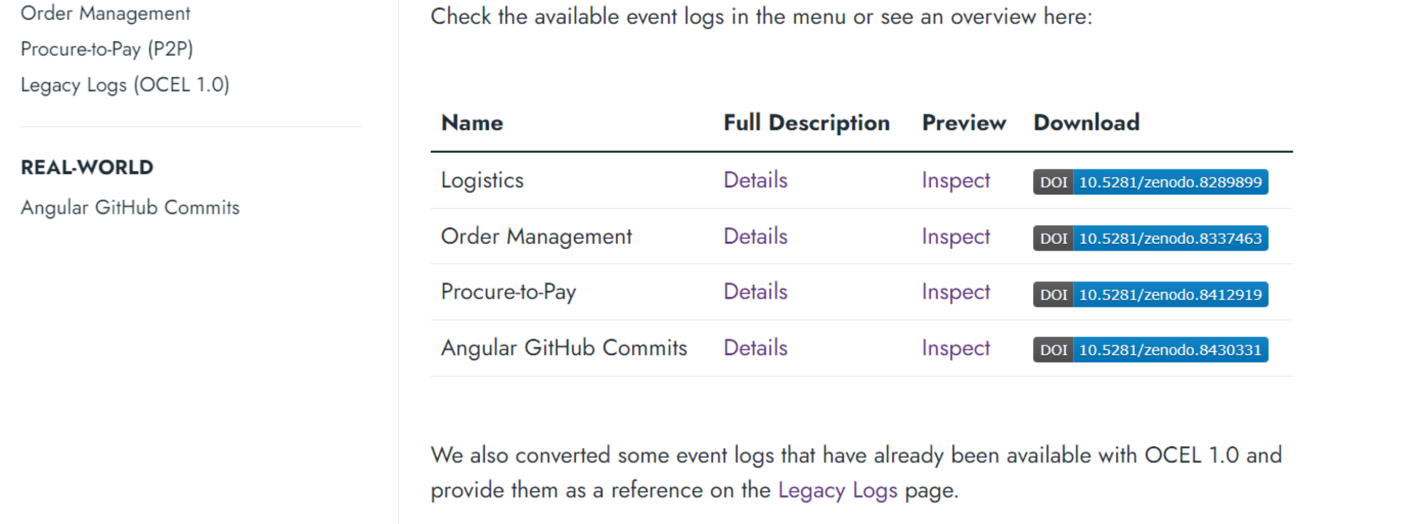


Ocelot — Object-Centric Event Log 2.0 Inspector

Object Types

☑ count object relations

☑ count co-occurrences in events



OCEL 2.0 Specification Event Logs Tool Support OCEL 1.0 Se

SIMULATIONS

- Logistics
- Order Management
- Procure-to-Pay (P2P)
- Legacy Logs (OCEL 1.0)

REAL-WORLD

- Angular GitHub Commits

Event Logs

Check the available event logs in the menu or see an overview here:

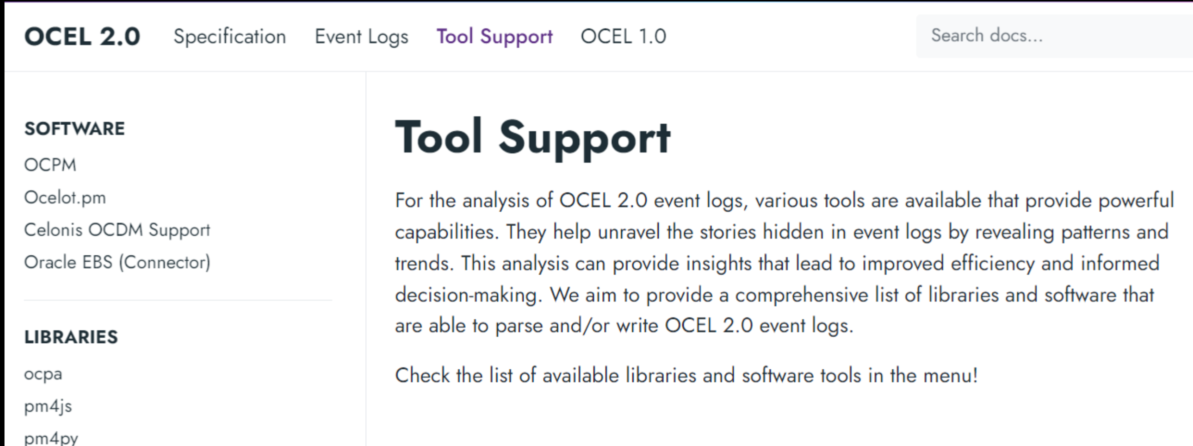
Name	Full Description	Preview	Download
Logistics	Details	Inspect	DOI 10.5281/zenodo.8289899
Order Management	Details	Inspect	DOI 10.5281/zenodo.8337463
Procure-to-Pay	Details	Inspect	DOI 10.5281/zenodo.8412919
Angular GitHub Commits	Details	Inspect	DOI 10.5281/zenodo.8430331

We also converted some event logs that have already been available with OCEL 1.0 and provide them as a reference on the Legacy Logs page.

© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

Tool support (New!)



SOFTWARE

- OCPM
- Ocelot.pm
- Celonis OCDM Support
- Oracle EBS (Connector)

LIBRARIES

- ocpa
- pm4js
- pm4py

EXAMPLES

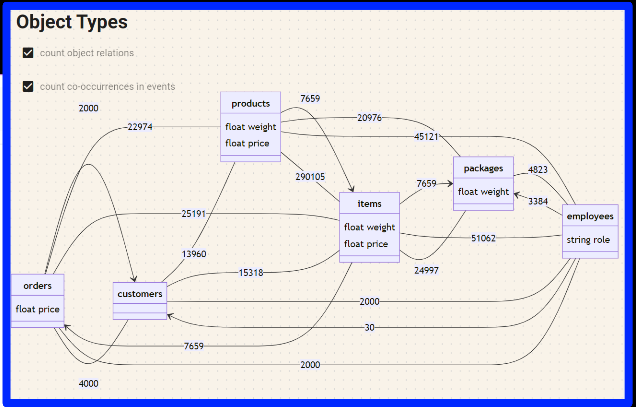
- Order Management

Tool Support

For the analysis of OCEL 2.0 event logs, various tools are available that provide powerful capabilities. They help unravel the stories hidden in event logs by revealing patterns and trends. This analysis can provide insights that lead to improved efficiency and informed decision-making. We aim to provide a comprehensive list of libraries and software that are able to parse and/or write OCEL 2.0 event logs.

Check the list of available libraries and software tools in the menu!

← Order Management

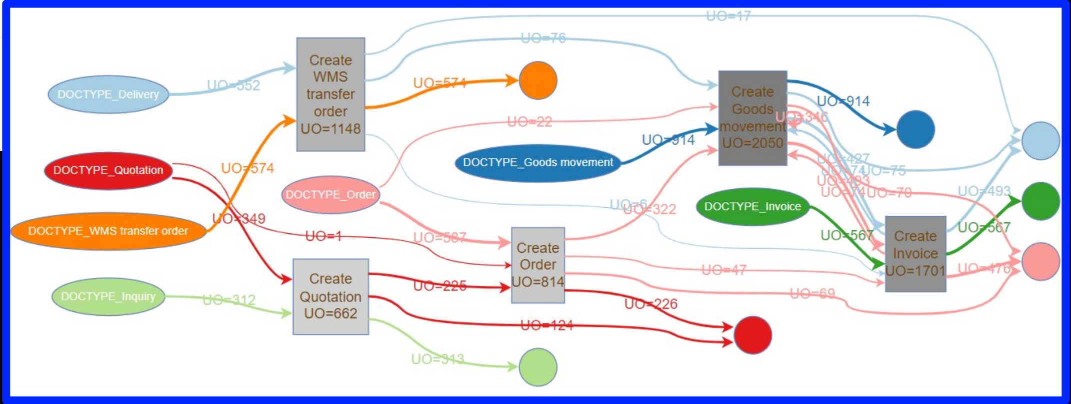


Object Types

☑ count object relations

☑ count co-occurrences in events

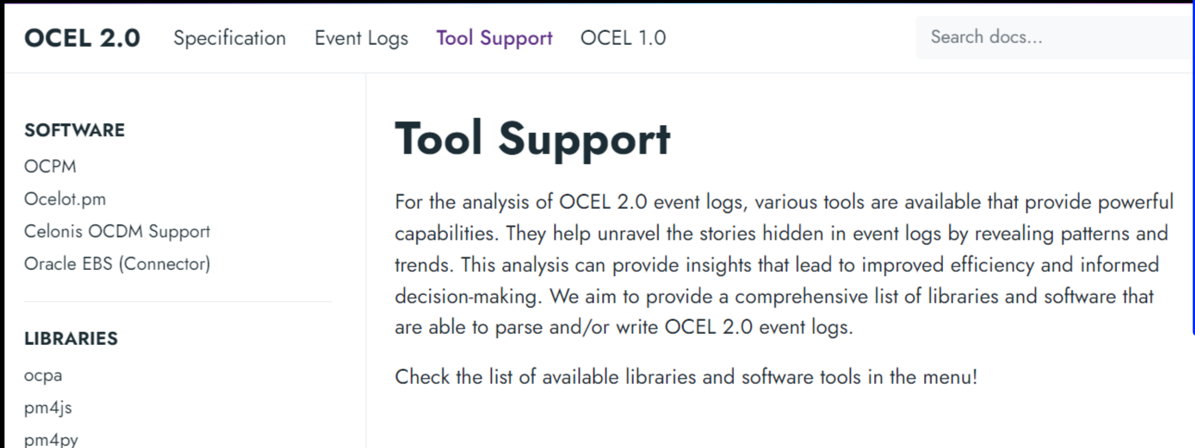
Event Log Export



© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

Tool support (New!)



OCEL 2.0 Specification Event Logs Tool Support OCEL 1.0 Search docs...

SOFTWARE

OCPM

Ocelot.pm

Celonis OCDM Support

Oracle EBS (Connector)

LIBRARIES

ocpa

pm4js

pm4py

EXAMPLES

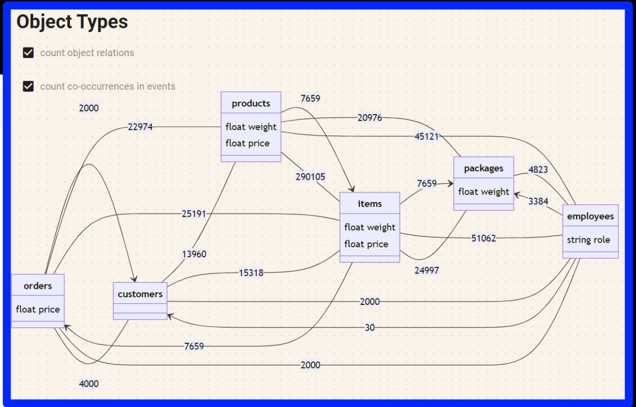
Order Management

Tool Support

For the analysis of OCEL 2.0 event logs, various tools are available that provide powerful capabilities. They help unravel the stories hidden in event logs by revealing patterns and trends. This analysis can provide insights that lead to improved efficiency and informed decision-making. We aim to provide a comprehensive list of libraries and software tools that are able to parse and/or write OCEL 2.0 event logs.

Check the list of available libraries and software tools in the menu!

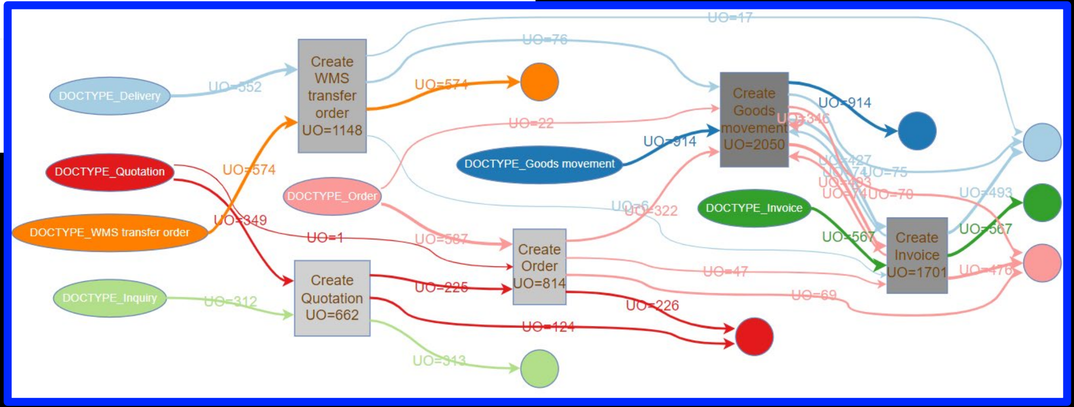
← Order Management



Object Types

☑ count object relations

☑ count co-occurrences in events



Event Log Export

</> EXPORT XML

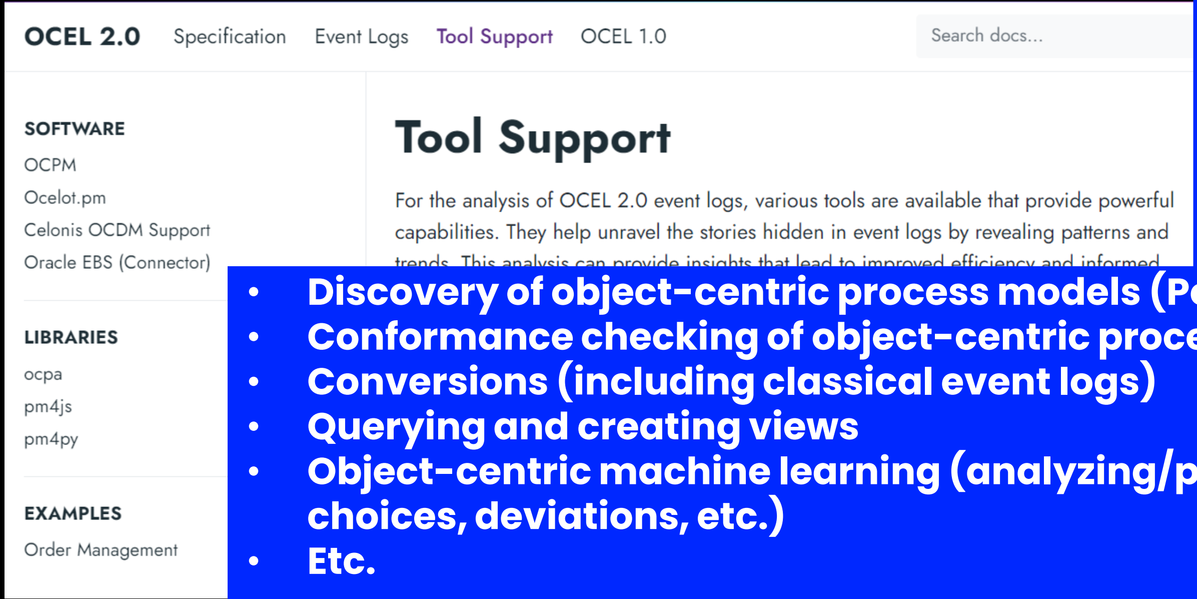
{..} EXPORT JSON

EXPORT SQLITE

© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

Tool support (New!)



OCEL 2.0 Specification Event Logs Tool Support OCEL 1.0

Search docs...

SOFTWARE

- OCPM
- Ocelot.pm
- Celonis OCDM Support
- Oracle EBS (Connector)

LIBRARIES

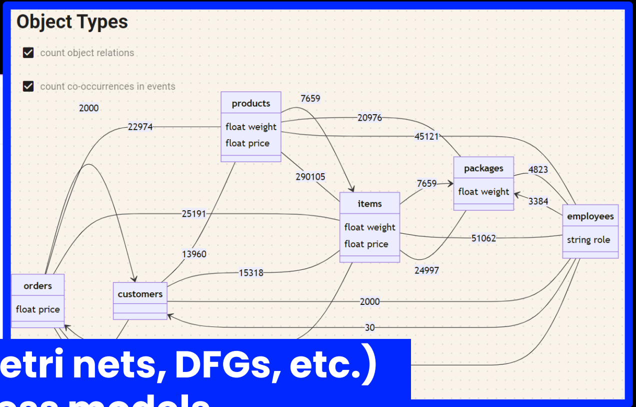
- ocpa
- pm4js
- pm4py

EXAMPLES

- Order Management

Tool Support

For the analysis of OCEL 2.0 event logs, various tools are available that provide powerful capabilities. They help unravel the stories hidden in event logs by revealing patterns and trends. This analysis can provide insights that lead to improved efficiency and informed



Object Types

☑ count object relations

☑ count co-occurrences in events

- Discovery of object-centric process models (Petri nets, DFGs, etc.)
- Conformance checking of object-centric process models
- Conversions (including classical event logs)
- Querying and creating views
- Object-centric machine learning (analyzing/predicting delays, choices, deviations, etc.)
- Etc.

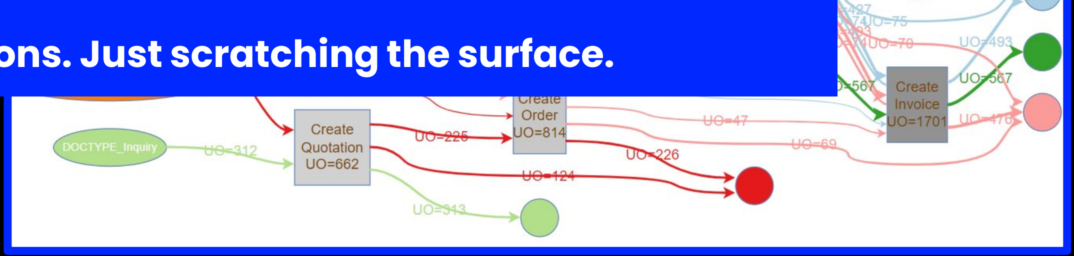
But also many open questions. Just scratching the surface.

Event Log Export

</> EXPORT XML

{..} EXPORT JSON

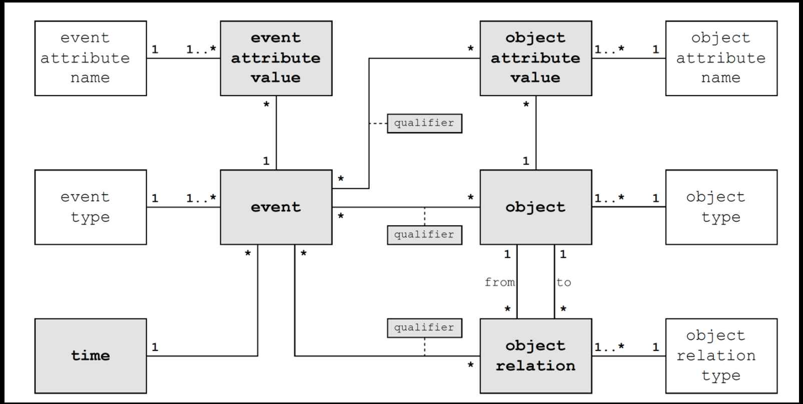
EXPORT SQLITE

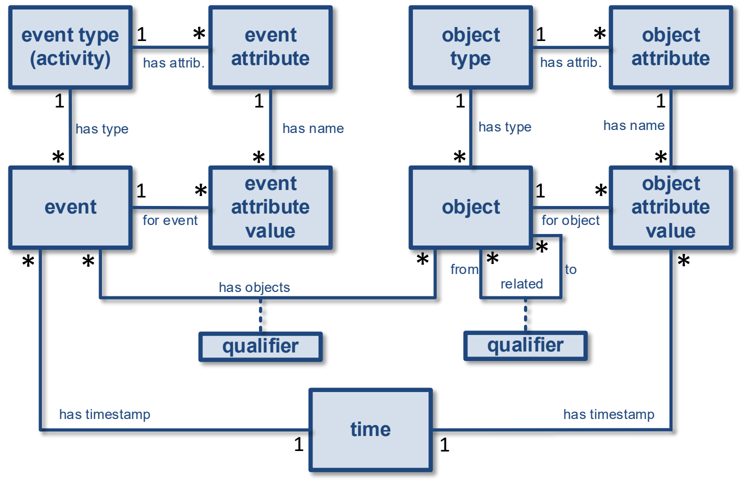


© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

Comparative analysis



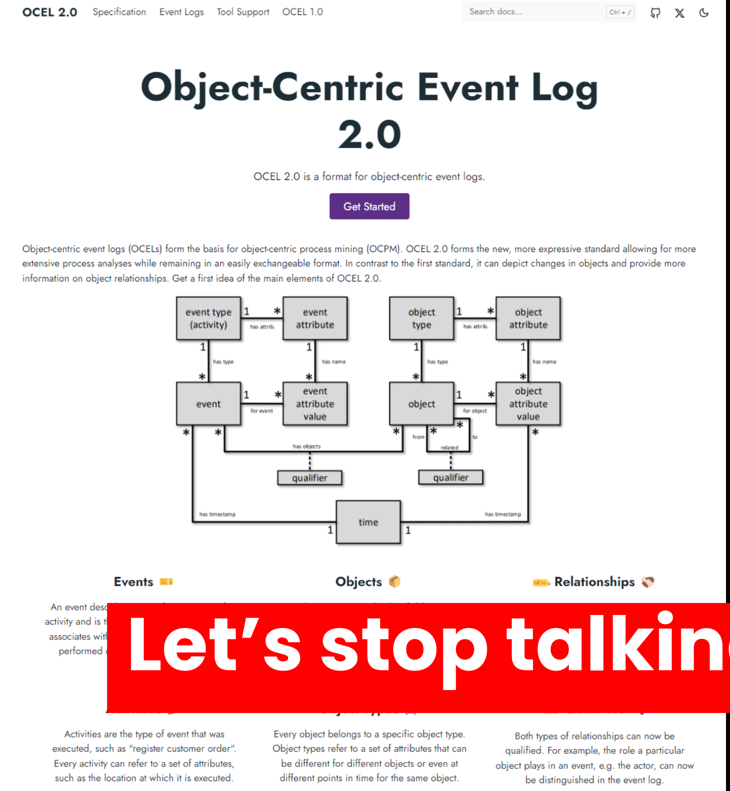


- Where to stop? Event-to-event, activity instances, etc.
- Shouldn't there be PM techniques first?
- Four ways of connecting events to objects?
- Object relations and object attribute values referring to multiple events?

- Object relations can easily be "objectified". This holds for many other extensions. "If you want to point to a relationship, turn it into an object."
- We define the set of expected attributes per type.
- We avoid having to create events for object attribute value changes (this is a design choice and allows for time series).

© Wil van der Aalst (use only with permission & acknowledgements)

Conclusion
OCEL 2.0: A new standard for OCED/OCPM



Object-Centric Event Log 2.0

OCEL 2.0 is a format for object-centric event logs.

Object-centric event logs (OCELs) form the basis for object-centric process mining (OCPM). OCEL 2.0 forms the new, more expressive standard allowing for more extensive process analyses while remaining in an easily exchangeable format. In contrast to the first standard, it can depict changes in objects and provide more information on object relationships. Get a first idea of the main elements of OCEL 2.0.

Events

An event desc
activity and is t
associates wit
performed

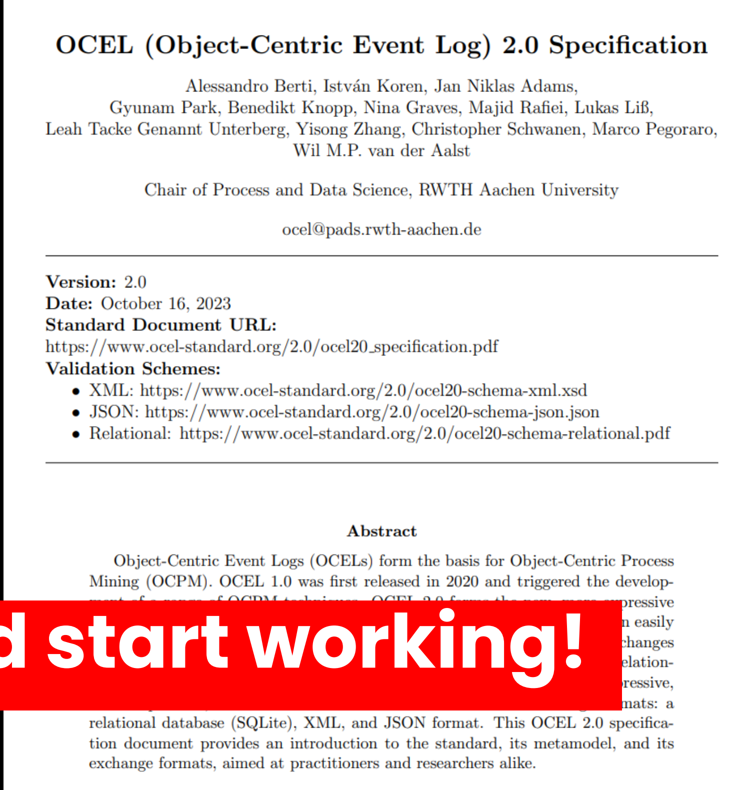
Objects

Relationships

Activities are the type of event that was executed, such as "register customer order". Every activity can refer to a set of attributes, such as the location at which it is executed.

Every object belongs to a specific object type. Object types refer to a set of attributes that can be different for different objects or even at different points in time for the same object.

Both types of relationships can now be qualified. For example, the role a particular object plays in an event, e.g. the actor, can now be distinguished in the event log.



OCEL (Object-Centric Event Log) 2.0 Specification

Alessandro Berti, István Koren, Jan Niklas Adams,
Gyunam Park, Benedikt Knopp, Nina Graves, Majid Rafiei, Lukas Liß,
Leah Tacke Genannt Unterberg, Yisong Zhang, Christopher Schwanen, Marco Pegoraro,
Wil M.P. van der Aalst

Chair of Process and Data Science, RWTH Aachen University

ocel@pads.rwth-aachen.de

Version: 2.0
Date: October 16, 2023
Standard Document URL:
https://www.ocel-standard.org/2.0/ocel20_specification.pdf
Validation Schemes:

- XML: https://www.ocel-standard.org/2.0/ocel20-schema-xml.xsd
- JSON: https://www.ocel-standard.org/2.0/ocel20-schema-json.json
- Relational: https://www.ocel-standard.org/2.0/ocel20-schema-relational.pdf

Abstract

Object-Centric Event Logs (OCELs) form the basis for Object-Centric Process Mining (OCPM). OCEL 1.0 was first released in 2020 and triggered the develop-
pressive
n easily
changes
elation-
ressive,
mats: a
relational database (SQLite), XML, and JSON format. This OCEL 2.0 specification document provides an introduction to the standard, its metamodel, and its exchange formats, aimed at practitioners and researchers alike.

Let's stop talking and start working!

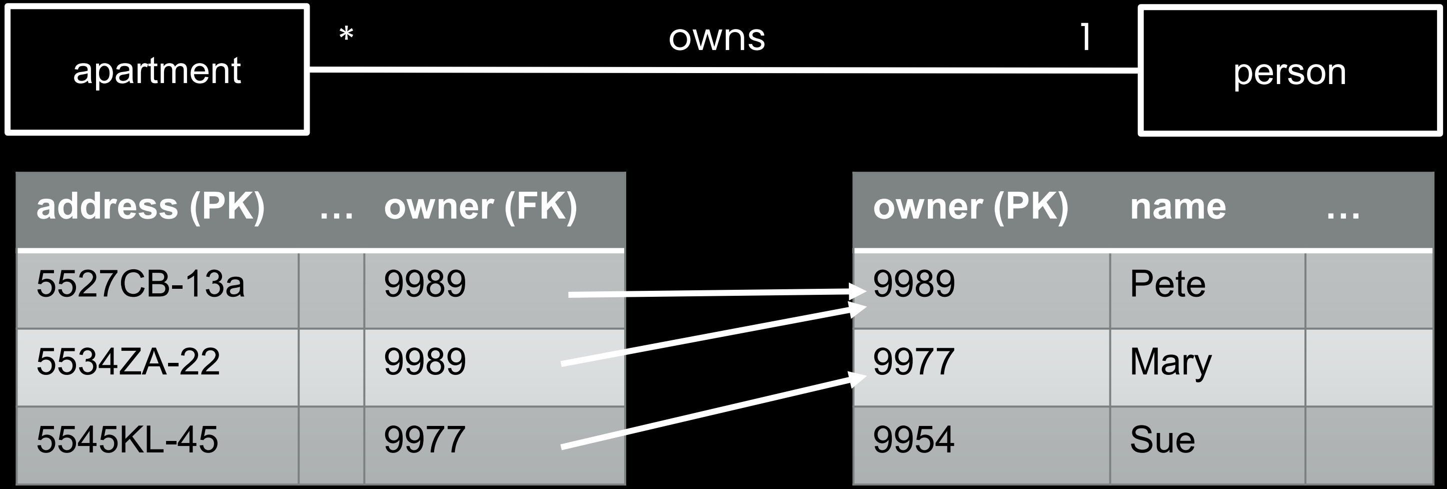
© Wil van der Aalst (use only with permission & acknowledgements)

https://www.ocel-standard.org/

Appendix

© Wil van der Aalst (use only with permission & acknowledgements)

Objectification

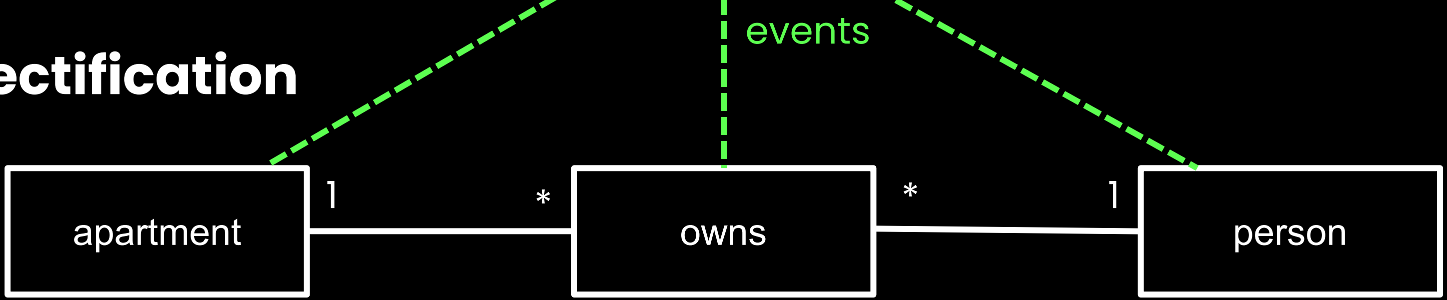


address (PK)	…	owner (FK)
5527CB-13a		9989
5534ZA-22		9989
5545KL-45		9977

owner (PK)	name	…
9989	Pete	
9977	Mary	
9954	Sue	

© Wil van der Aalst (use only with permission & acknowledgements)

Objectification

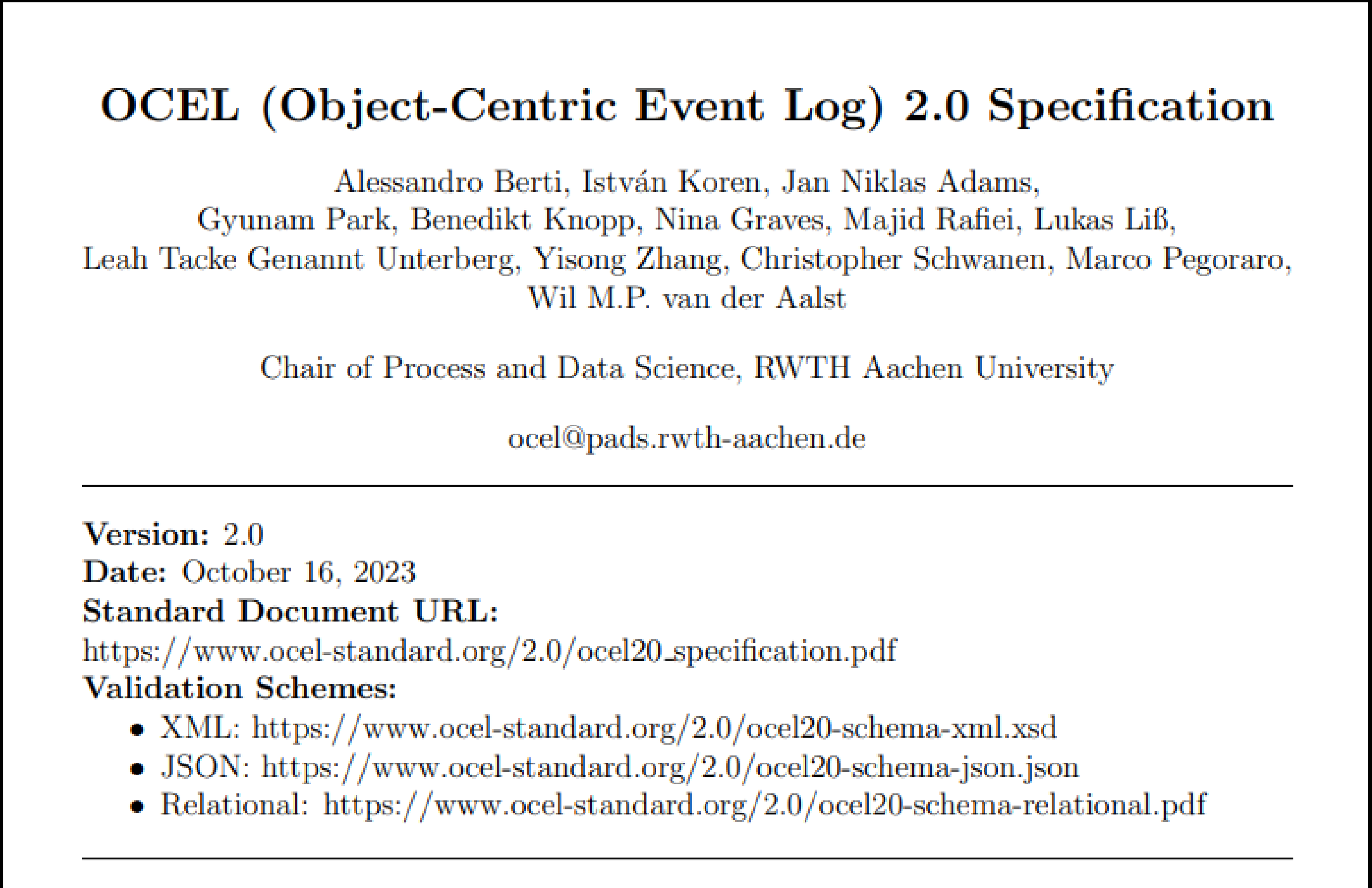


address (PK)	…
5527CB-13a	…
5534ZA-22	…
5545KL-45	…

owner (PK)	name	…
9989	Pete	…
9977	Mary	…
9954	Sue	…

address	owner	contract (PK)	…
5527CB-13a	9989	con145	…
5527CB-13a	9977	con457	…
5534ZA-22	9989	con569	…
5545KL-45	9977	con679	..

© Wil van der Aalst (use only with permission & acknowledgements)



OCEL (Object-Centric Event Log) 2.0 Specification

Alessandro Berti, István Koren, Jan Niklas Adams,
Gyunam Park, Benedikt Knopp, Nina Graves, Majid Rafiei, Lukas Liß,
Leah Tacke Genannt Unterberg, Yisong Zhang, Christopher Schwanen, Marco Pegoraro,
Wil M.P. van der Aalst

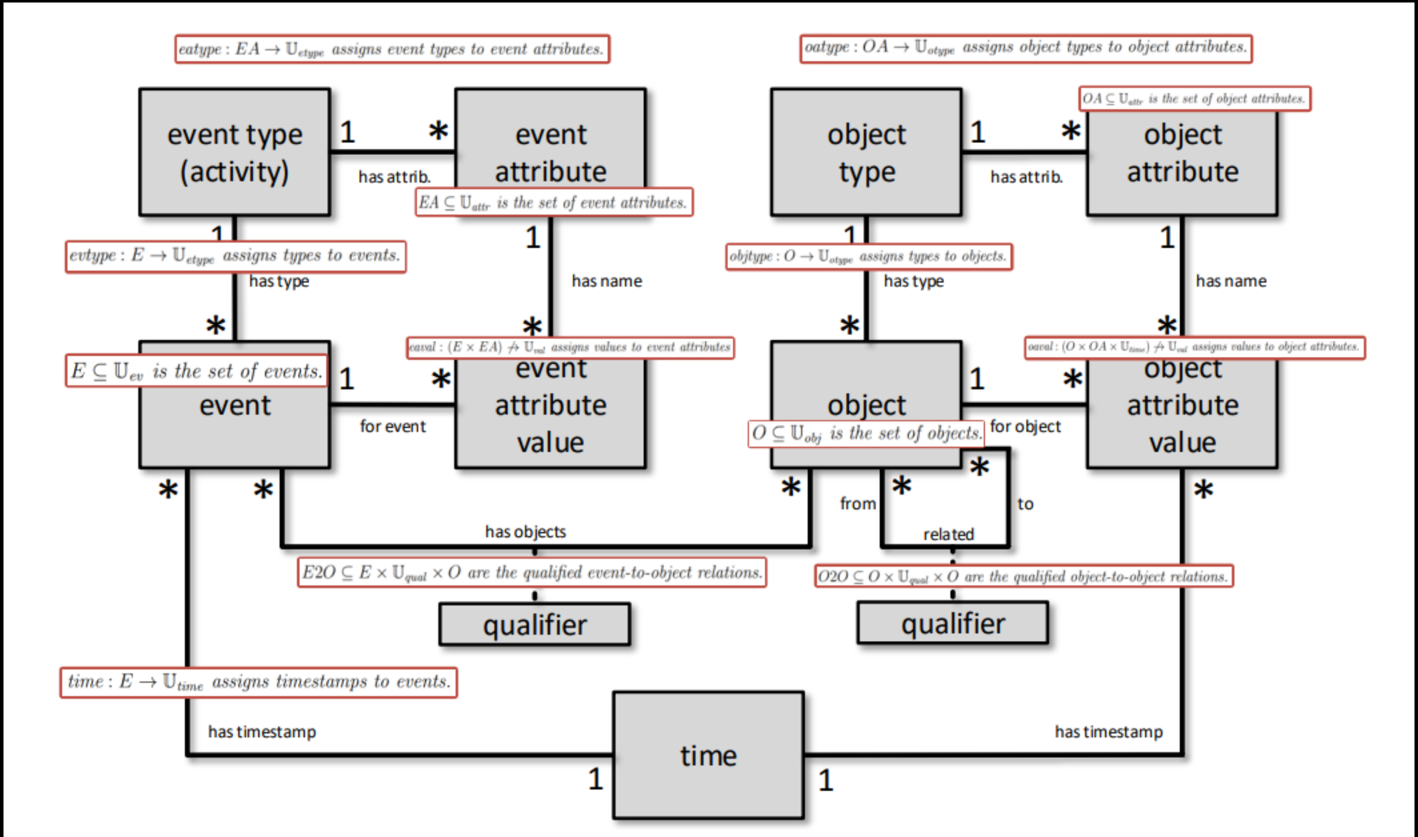
Chair of Process and Data Science, RWTH Aachen University

ocel@pads.rwth-aachen.de

Version: 2.0
Date: October 16, 2023
Standard Document URL:
https://www.ocel-standard.org/2.0/ocel20_specification.pdf
Validation Schemes:

- XML: https://www.ocel-standard.org/2.0/ocel20-schema-xml.xsd
- JSON: https://www.ocel-standard.org/2.0/ocel20-schema-json.json
- Relational: https://www.ocel-standard.org/2.0/ocel20-schema-relational.pdf

© Wil van der Aalst (use only with permission & acknowledgements)



$eatype : EA \rightarrow \mathbb{U}_{etype}$ *assigns event types to event attributes.*

$oatype : OA \rightarrow \mathbb{U}_{otype}$ *assigns object types to object attributes.*

$OA \subseteq \mathbb{U}_{attr}$ *is the set of object attributes.*

$EA \subseteq \mathbb{U}_{attr}$ *is the set of event attributes.*

$evtype : E \rightarrow \mathbb{U}_{etype}$ *assigns types to events.*

$objtype : O \rightarrow \mathbb{U}_{otype}$ *assigns types to objects.*

$eaval : (E \times EA) \not\rightarrow \mathbb{U}_{val}$ *assigns values to event attributes*

$oaval : (O \times OA \times \mathbb{U}_{time}) \not\rightarrow \mathbb{U}_{val}$ *assigns values to object attributes.*

$E \subseteq \mathbb{U}_{ev}$ *is the set of events.*

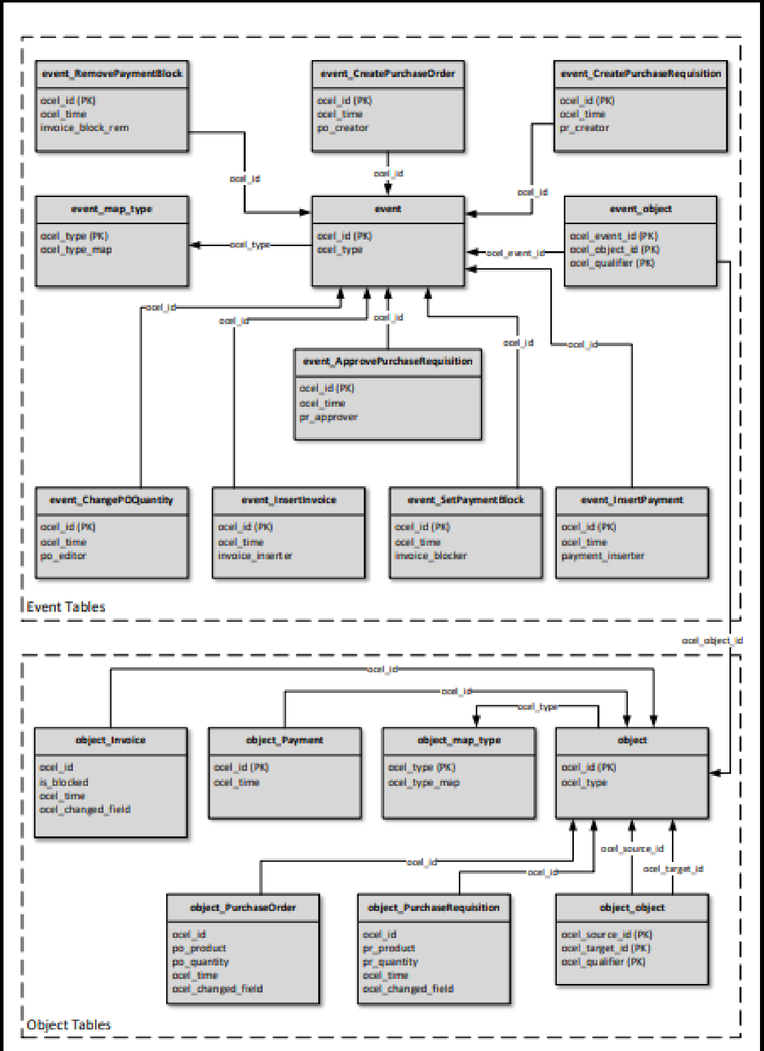
$O \subseteq \mathbb{U}_{obj}$ *is the set of objects.*

$E2O \subseteq E \times \mathbb{U}_{qual} \times O$ *are the qualified event-to-object relations.*

$O2O \subseteq O \times \mathbb{U}_{qual} \times O$ *are the qualified object-to-object relations.*

$time : E \rightarrow \mathbb{U}_{time}$ *assigns timestamps to events.*

© Wil van der Aalst (use only with permission & acknowledgements)



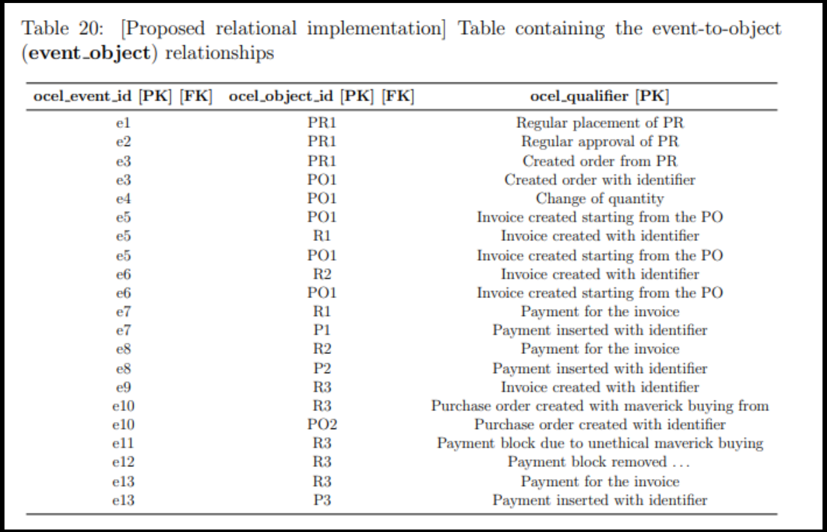


Table 20: [Proposed relational implementation] Table containing the event-to-object (**event_object**) relationships

ocel_event_id [PK] [FK]	ocel_object_id [PK] [FK]	ocel_qualifier [PK]
e1	PR1	Regular placement of PR
e2	PR1	Regular approval of PR
e3	PR1	Created order from PR
e3	PO1	Created order with identifier
e4	PO1	Change of quantity
e5	PO1	Invoice created starting from the PO
e5	R1	Invoice created with identifier
e5	PO1	Invoice created starting from the PO
e6	R2	Invoice created with identifier
e6	PO1	Invoice created starting from the PO
e7	R1	Payment for the invoice
e7	P1	Payment inserted with identifier
e8	R2	Payment for the invoice
e8	P2	Payment inserted with identifier
e9	R3	Invoice created with identifier
e10	R3	Purchase order created with maverick buying from
e10	PO2	Purchase order created with identifier
e11	R3	Payment block due to unethical maverick buying
e12	R3	Payment block removed . . .
e13	R3	Payment for the invoice
e13	P3	Payment inserted with identifier

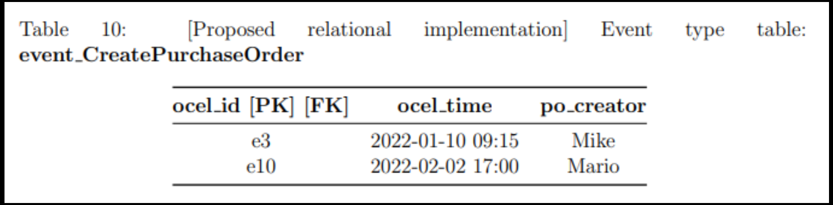


Table 10: [Proposed relational implementation] Event type table: **event_CreatePurchaseOrder**

ocel_id [PK] [FK]	ocel_time	po_creator
e3	2022-01-10 09:15	Mike
e10	2022-02-02 17:00	Mario

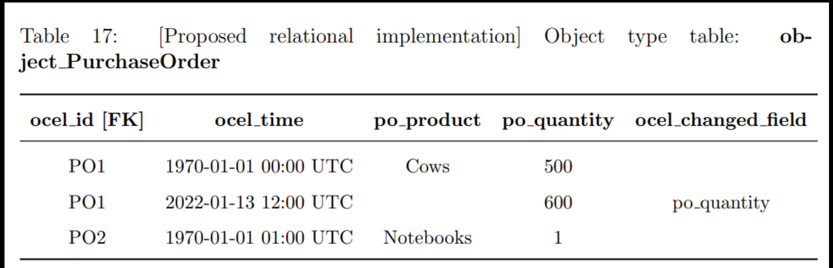
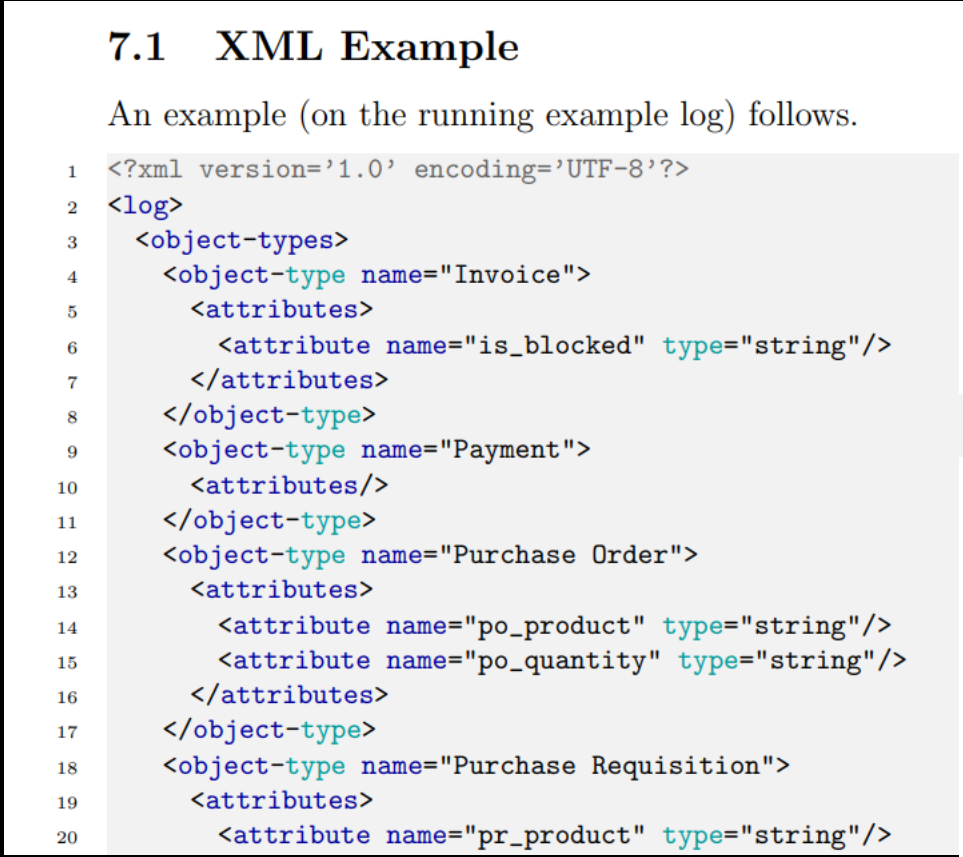


Table 17: [Proposed relational implementation] Object type table: **object_PurchaseOrder**

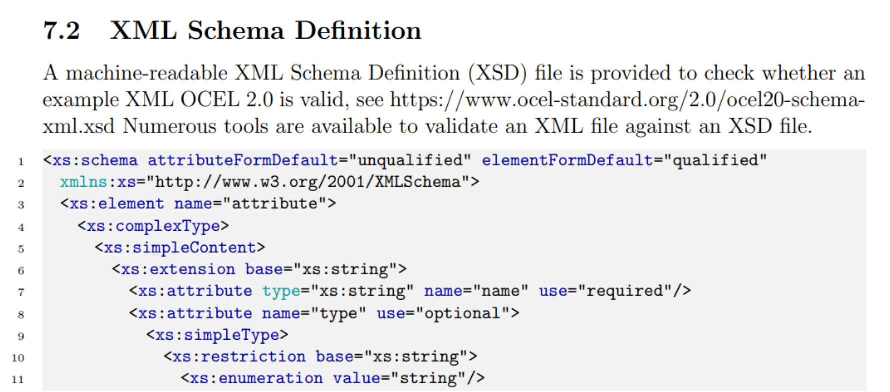
ocel_id [FK]	ocel_time	po_product	po_quantity	ocel_changed_field
PO1	1970-01-01 00:00 UTC	Cows	500	
PO1	2022-01-13 12:00 UTC		600	po_quantity
PO2	1970-01-01 01:00 UTC	Notebooks	1	



7.1 XML Example

An example (on the running example log) follows.

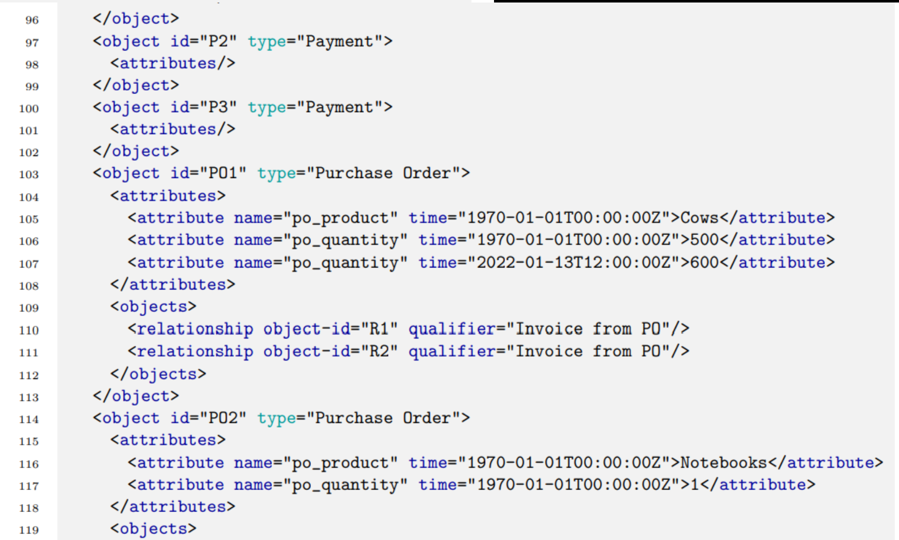
```
<?xml version=’1.0’ encoding=’UTF-8’?>
<log>
  <object-types>
    <object-type name="Invoice">
      <attributes>
        <attribute name="is_blocked" type="string"/>
      </attributes>
    </object-type>
    <object-type name="Payment">
      <attributes/>
    </object-type>
    <object-type name="Purchase Order">
      <attributes>
        <attribute name="po_product" type="string"/>
        <attribute name="po_quantity" type="string"/>
      </attributes>
    </object-type>
    <object-type name="Purchase Requisition">
      <attributes>
        <attribute name="pr_product" type="string"/>
```



7.2 XML Schema Definition

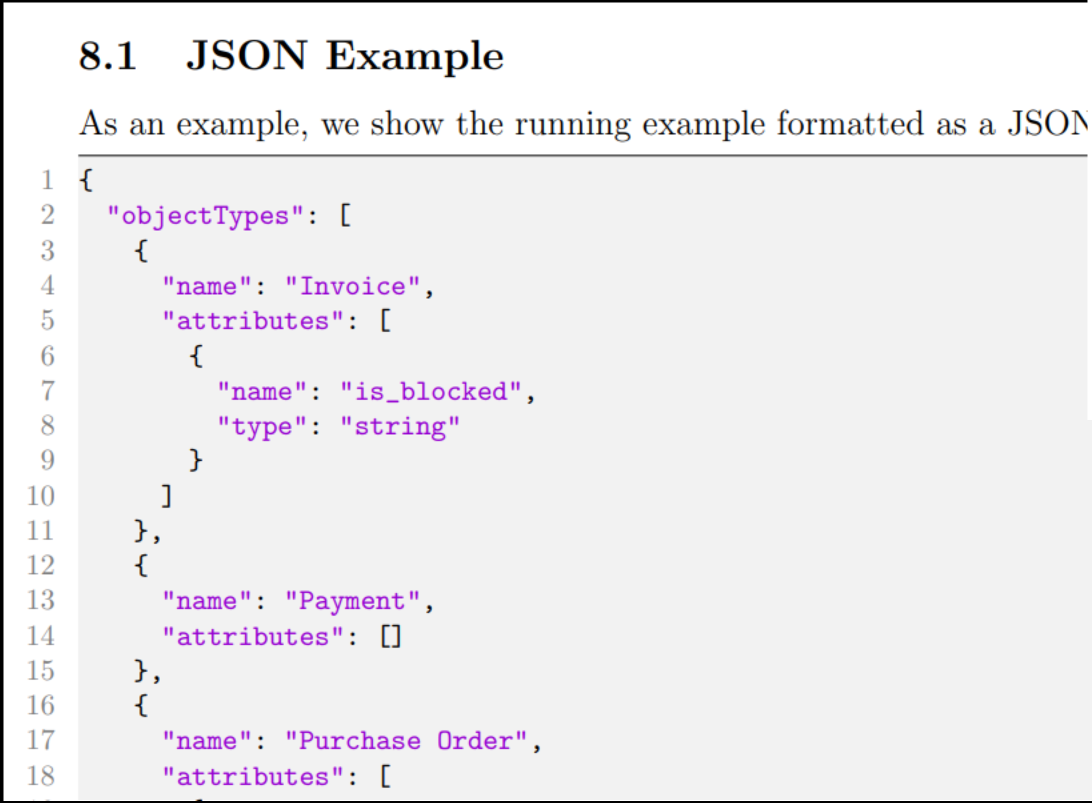
A machine-readable XML Schema Definition (XSD) file is provided to check whether an example XML OCEL 2.0 is valid, see https://www.ocel-standard.org/2.0/ocel20-schema-xml.xsd Numerous tools are available to validate an XML file against an XSD file.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
 xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="attribute">
   <xs:complexType>
     <xs:simpleContent>
       <xs:extension base="xs:string">
         <xs:attribute type="xs:string" name="name" use="required"/>
         <xs:attribute name="type" use="optional">
           <xs:simpleType>
             <xs:restriction base="xs:string">
               <xs:enumeration value="string"/>
```



```
    </object>
    <object id="P2" type="Payment">
      <attributes/>
    </object>
    <object id="P3" type="Payment">
      <attributes/>
    </object>
    <object id="PO1" type="Purchase Order">
      <attributes>
        <attribute name="po_product" time="1970-01-01T00:00:00Z">Cows</attribute>
        <attribute name="po_quantity" time="1970-01-01T00:00:00Z">500</attribute>
        <attribute name="po_quantity" time="2022-01-13T12:00:00Z">600</attribute>
      </attributes>
      <objects>
        <relationship object-id="R1" qualifier="Invoice from PO"/>
        <relationship object-id="R2" qualifier="Invoice from PO"/>
      </objects>
    </object>
    <object id="PO2" type="Purchase Order">
      <attributes>
        <attribute name="po_product" time="1970-01-01T00:00:00Z">Notebooks</attribute>
        <attribute name="po_quantity" time="1970-01-01T00:00:00Z">1</attribute>
      </attributes>
      <objects>
```

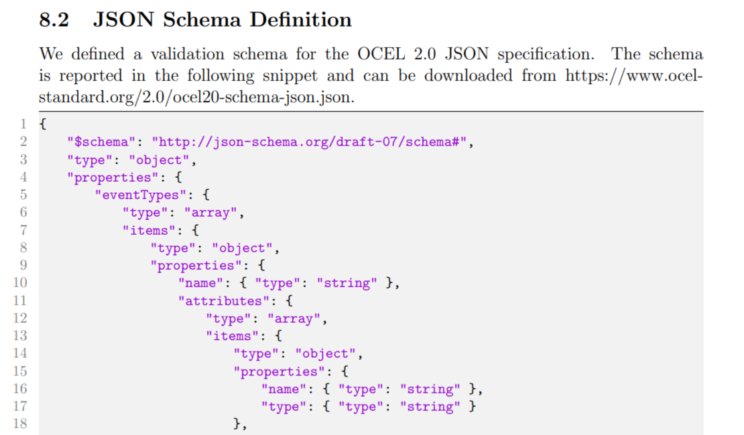
© Wil van der Aalst (use only with permission & acknowledgements)



8.1 JSON Example

As an example, we show the running example formatted as a JSON

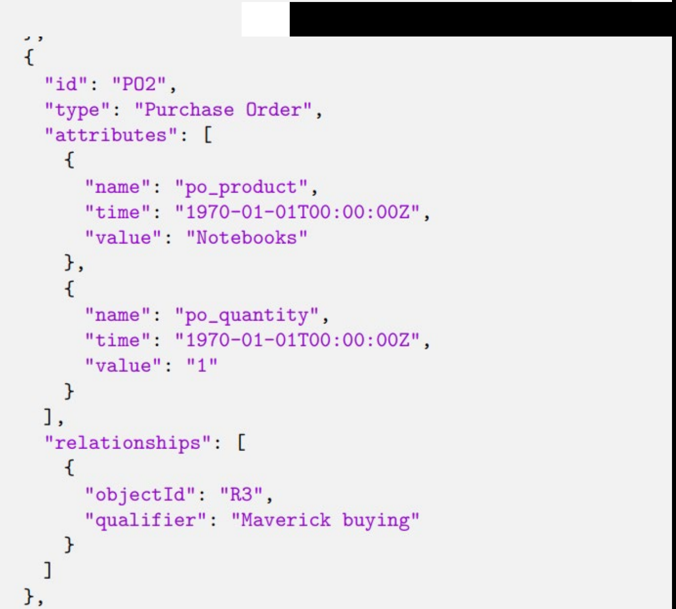
```
{
  "objectTypes": [
    {
      "name": "Invoice",
      "attributes": [
        {
          "name": "is_blocked",
          "type": "string"
        }
      ]
    },
    {
      "name": "Payment",
      "attributes": []
    },
    {
      "name": "Purchase Order",
      "attributes": [
```



8.2 JSON Schema Definition

We defined a validation schema for the OCEL 2.0 JSON specification. The schema is reported in the following snippet and can be downloaded from https://www.ocel-standard.org/2.0/ocel20-schema-json.json.

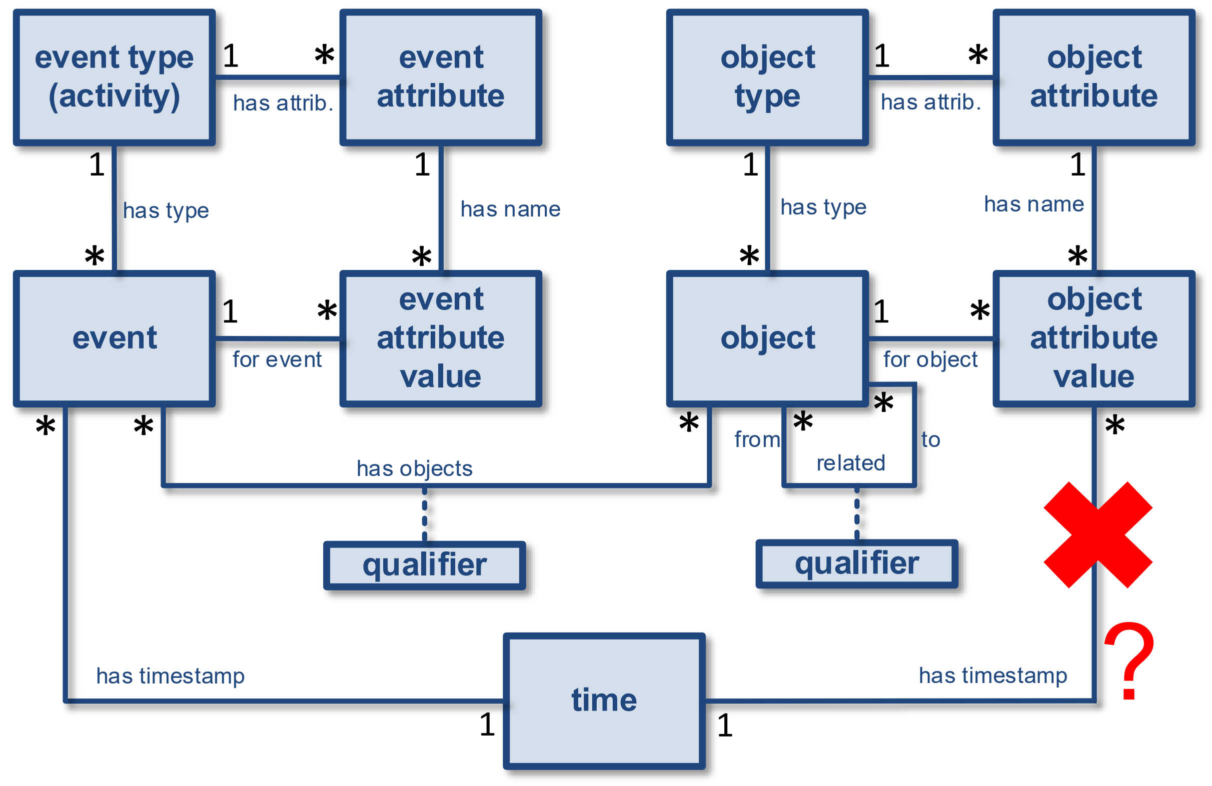
```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "type": "object",
    "properties": {
        "eventTypes": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "name": { "type": "string" },
                    "attributes": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
                                "name": { "type": "string" },
                                "type": { "type": "string" }
                            },
```



```
  },
  {
    "id": "PO2",
    "type": "Purchase Order",
    "attributes": [
      {
        "name": "po_product",
        "time": "1970-01-01T00:00:00Z",
        "value": "Notebooks"
      },
      {
        "name": "po_quantity",
        "time": "1970-01-01T00:00:00Z",
        "value": "1"
      }
    ],
    "relationships": [
      {
        "objectId": "R3",
        "qualifier": "Maverick buying"
      }
    ]
  },
```

© Wil van der Aalst (use only with permission & acknowledgements)

Greatest Common Divisor (GCD) ?



© Wil van der Aalst (use only with permission & acknowledgements)

www.ocel-standard.org

© Wil van der Aalst (use only with permission & acknowledgements)