

XES

Reflections and Recommendations

prof.dr.ir. Wil van der Aalst
RWTH Aachen University
W: vdaalst.com T: @wvdaalst

IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams

IEEE Computational Intelligence Society

Sponsored by the
Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1849™-2016

Adopted as an IEEE standard in 2016, but this started in 2009 with a first draft of XES as input for the TFPM.

From the Minutes of the Meeting of the IEEE Task Force on Process Mining, September 15th 2010, Stevens Institute of Technology, Hoboken NJ: "... All participants agree with using the format in the future. ... The task force adopts the XES standard and will use it as a tool to promote the topic."

Participants

At the time this IEEE standard was completed, the XES Working Group had the following membership:

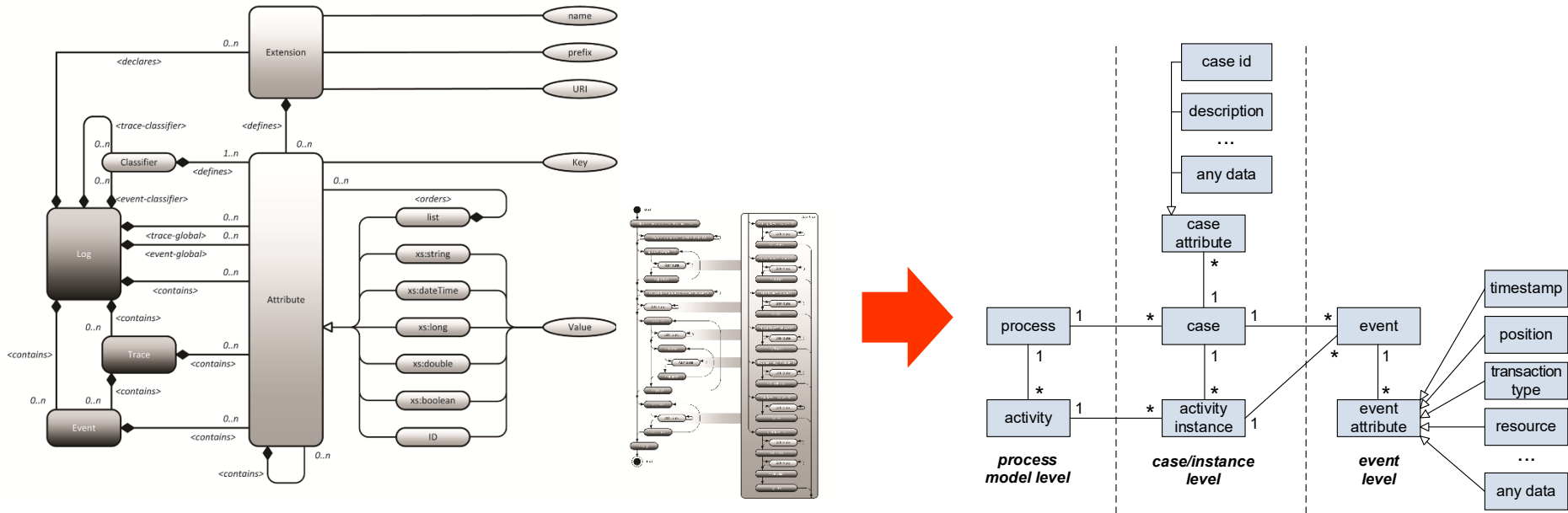
Wil van der Aalst, *Chair*
Christian Günther, *Vice Chair*

JC Bose	Teemu Lehto	Anne Rozinat
Josep Carmona	Felix Mannhardt	Pinna Soffer
Marlon Dumas	Marco Montali	Minseok Song
Frank van Geffen	Michael zur Muehlen	Keith Swenson
Sukriti Goel	Zbigniew Paszkiewicz	Walter Vanherle
Antonella Guzzo	Hajo Reijers	Eric Verbeek
Rania Khalaf	Alexander Rinke	Lijie Wen
Rudolf Kuhn	Michal Rosik	Moe Wynn

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Giovanni Acampora	Werner Holz	Zbigniew Paszkiewicz
Michael Bauer	Noriyuki Ikeuchi	Michal Rosik
JC Bose	Piotr Karocki	Christopher Turner
Keith Chow	Akhil Kumar	Wil van der Aalst
Randall Groves	Edward McCall	Eric Verbeek
Christian Günther	Björn Molitor	Lijie Wen
Antonella Guzzo	Takahide Nogayama	Oren Yuen

Focus on XES syntax was unintended

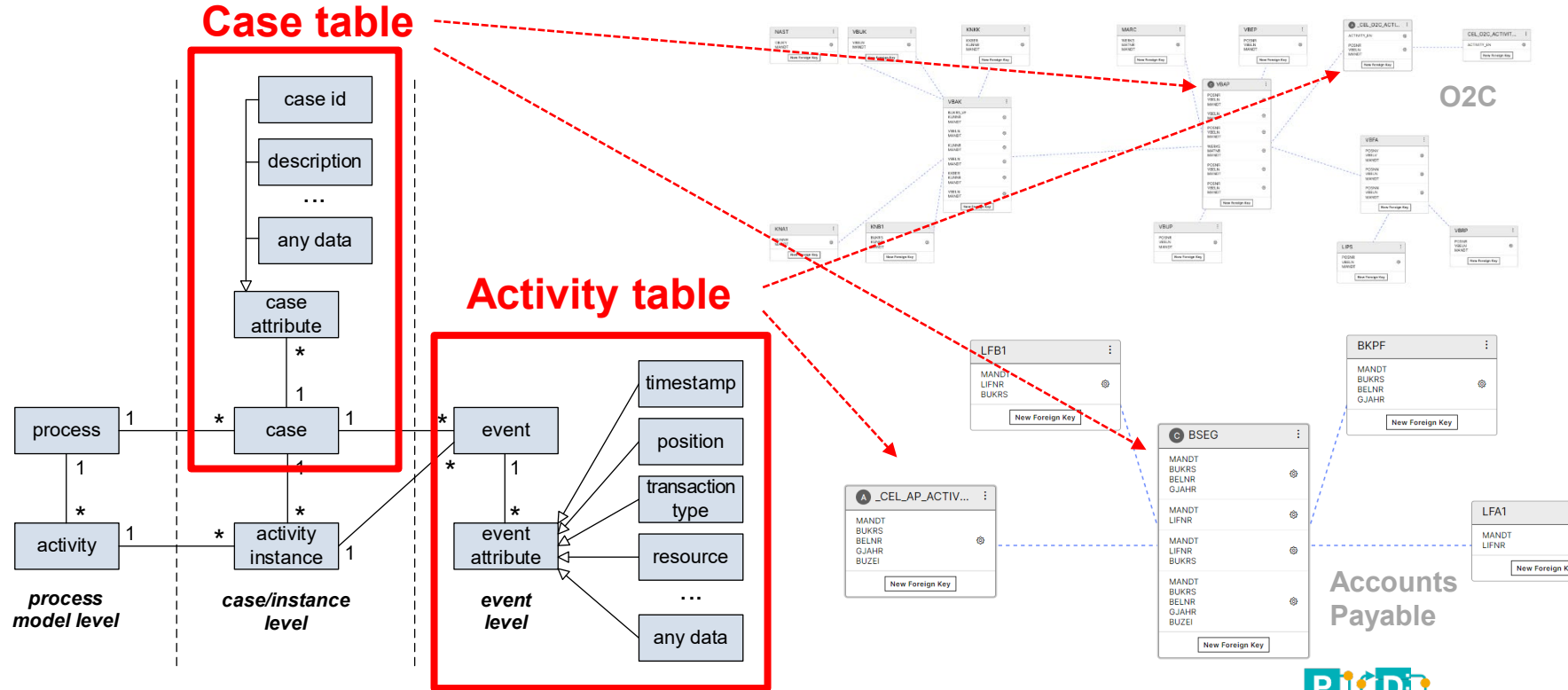


In hindsight a few things were probably over-engineered.

Diagram in process mining book.

See also alternative XES storage formats, e.g., JXES (JSON-based XES).

Example realization in Celonis



We need more “event producers”

Tools supporting XES

Tool name	Version	Vendor	
Apromore	7.16	Apromore	Import, Export: A1, B1, C1, D1 (PDF , 7,462 KB)
Celonis Process Mining	4.2	Celonis SE	Import: A1, B1, C1, D1 (PDF , 9,277 KB)
Disco	2.9	Fluxicon	Import, Export: A-D, X (PDF , 19,746 KB)
Icris Process Mining Factory	1	Icris	
MEHRWERK Process Mining	V2	MEHRWERK GmbH	Import, Export: A-D, X (PDF , 21,515 KB)
Minit	1	Gradient ECM	
myInvenio	1	Cognitive Technology	
PM4KNIME	1.0.0	Open Source by PADS, RWTH	Import, Export: A-D, X (PDF , 2,202 KB)
Pm4py	0.1	Open Source by PADS, RWTH	Import, Export: A-D, X (PDF , 3,006 KB)
PMLab-Lite	0.4.5	Community, including Polytechnic University of Catalonia, Technical University of Denmark, Humboldt-Universität zu Berlin	Import, Export: A-D, X (PDF , 2,857 KB)
Process Diamond Intelligence	1.6.0708	Process Diamond Pty Ltd	Import, Export: A1, B1, C1, D1, X1 (PDF , 8,607 KB)
ProcessGold Enterprise Platform	14	Processgold International B.V.	Import: A-D, X (PDF , 6,109 KB)
ProM	6.7	Open Source hosted at TU/e	Import, Export: A-D, X (PDF , 41,447 KB)
ProM Lite	1.2	Open Source hosted at TU/e	
QPR ProcessAnalyzer	2018.6	QPR	Import: A1, B1, C1, D1 (PDF , 6,109 KB)
RapidProM	4.0.1	Open Source by PADS, RWTH and TU/e	Import, Export: A-D (PDF , 2,426 KB)
Rialto Process	1.5	Exeura	
SNP Business Process Analysis	15.27	SNP Schneider-Neureither & Partner AG	

- Why are not all vendors supporting this although it is 2 days of work?
 - End-users are not demanding this.
 - Most XES data sets are for scientific competitions.
- Game will change when ERP/CRM vendors start to export event data in XES format.
 - How to make that happen?

Activity instances: Important, but rarely used (not logged and not used in tools)

7.1 Concept extension

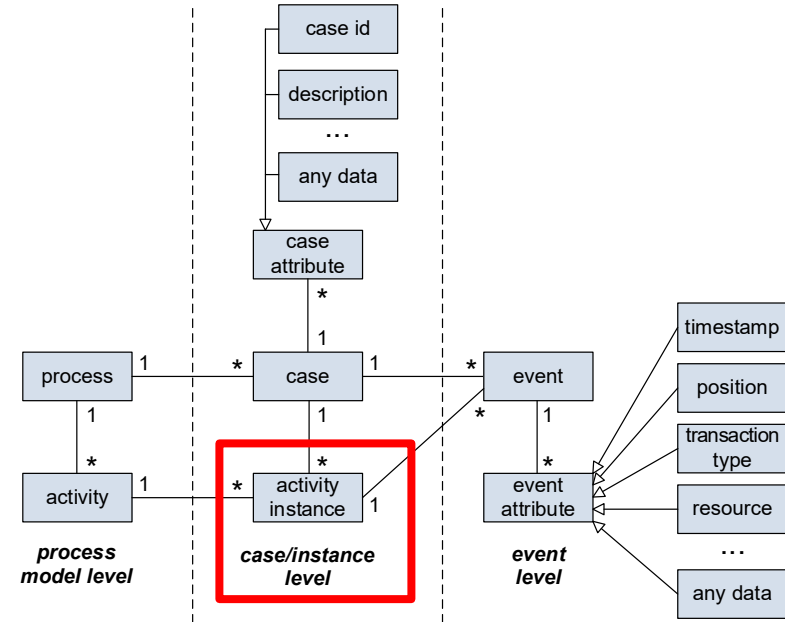
The Concept extension defines, for all levels of the XES type hierarchy, an attribute which stores the generally understood name of type hierarchy elements.

Name	Prefix	URI
Concept	concept	http://www.xes-standard.org/concept.xesext

7.1.1 Attributes

Name	Key	Components	Datatype	Description
Name	name	Log, Trace, Event	xs:string	Stores a generally understood name for any component type. For streams and logs, the name attribute may store the name of the process having been executed. For traces, the name attribute usually stores the case ID. For events, the name attribute represents the name of the event, e.g., the name of the executed activity represented by the event.
Instance	instance	Event	xs:string	This represents an identifier of the activity instance whose execution has generated the event. This way, multiple instances (occurrences) of the same activity can be told apart.

Instance: “*This represents an identifier of the activity instance whose execution has generated the event. This way, multiple instances (occurrences) of the same activity can be told apart.*”



Cost Extension: Also rarely used (not logged and not used in tools)

7.7 Cost extension

The cost extension defines a nested element to store information about the cost associated with activities within a log. The objective of this extension is to provide semantics to cost aspects that can be associated with events in a log. The definition associates three data elements with a particular cost element: the amount associated with the cost element as well as the cost driver that is responsible for incurring that cost and the cost type. As it is possible for more than one cost element to be associated with an event, the cost incurred per event is summarized using the total attribute. The currency element is also recorded once per event. Cost information can be recorded at the trace level (for instance, to be able to say that it costs \$20 when a case is started). Cost information can also be recorded at the event level (for instance, for certain event types such as complete or canceled events) to capture the cost incurred in undertaking the activity by a resource.

Name	Prefix	URI
Cost	cost	http://www.xes-standard.org/cost.xesext

7.7.1 Attributes

Name	Key	Components	Datatype	Description
Total	total	Trace, Event	xs:double	Total cost incurred for a trace or an event. The value represents the sum of all the cost amounts within the element.
Currency	currency	Trace, Event	xs:string	The currency (using the ISO 4217:2008 standard) of all costs of this element.
Drivers	drivers	Trace, Event	List	A detailed list containing cost driver details.

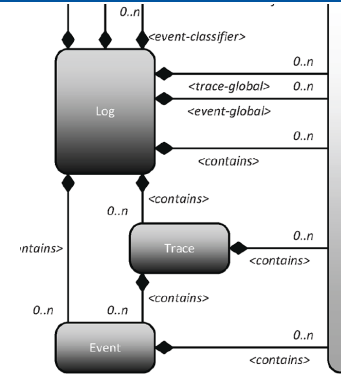
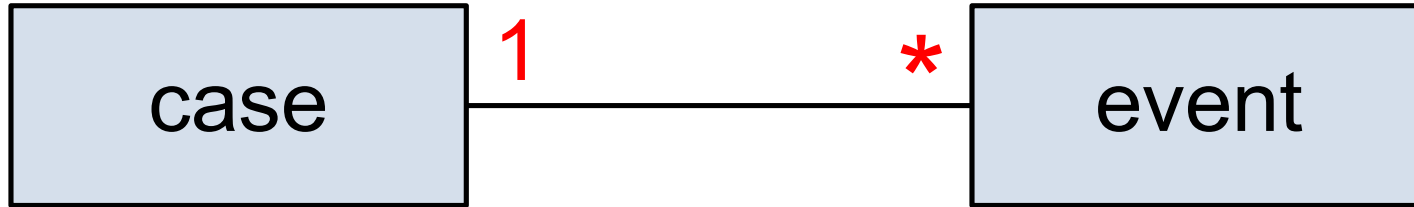
Name	Key	Components	Datatype	Description
Amount	amount	Meta	xs:double	The value contains the cost amount for a cost driver.
Driver	driver	Meta	xs:string	The value contains the id for the cost driver.
Type	type	Meta	xs:string	The value contains the cost type (e.g., Fixed, Overhead, Materials).

The drivers attribute shall contain any number of driver attributes, and every driver attribute shall contain the amount and type attribute, like follows:

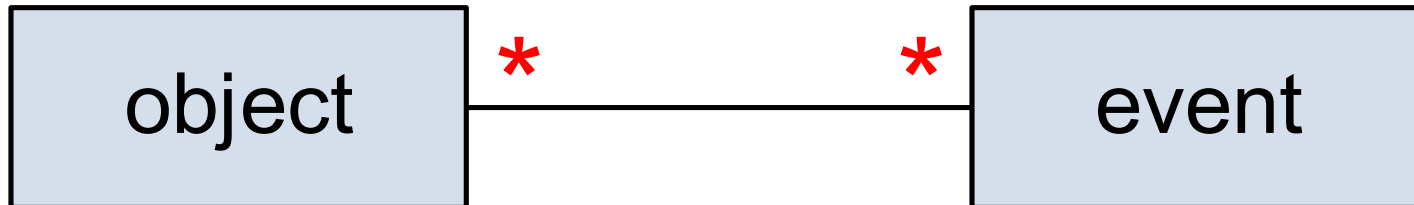
```
<event>
  <string key="cost:currency" value="AUD" />
  <string key="cost:total" value="123.50" />
  <list key="cost:drivers">
    <values>
      <string key="driver" value="d2f4ee27">
        <float key="amount" value="21.40" />
        <string key="type" value="Labour" />
      </string>
      <string key="driver" value="abc124">
        <float key="amount" value="102.10" />
        <string key="type" value="Variable Overhead" />
      </string>
    </values>
  </list>
</event>
```

Yet, process mining dashboards show financial results all the time!

Missing in XES: Object-centricity



Each event refers to a single case and events are only related through cases!



event = activity + timestamp + objects + attributes

Convergence and Divergence



- If **item** is the case id, then activities at the **order** and **package** level get replicated (**convergence problem**).
 - `place_order (order1,item1,item2,item3)` becomes `place_order(item1)`, `place_order(item2)`, `place_order(item3)`
- If **order** is the case id, then causal relations at the level of **items** get lost (**divergence problem**).
 - `pick_item (order1,item1)`, `pick_item (order1,item2)`, `pack_item (order1,item1)`, `pack_item (order1,item2)`, becomes `pick_item (order1)`, `pick_item (order1)`, `pack_item (order1)`, `pack_item (order1)`





OCEL STANDARD

[Introduction](#)[Format](#)[Event Logs](#)[Tool Support](#)[Contact](#)[Download Document](#)

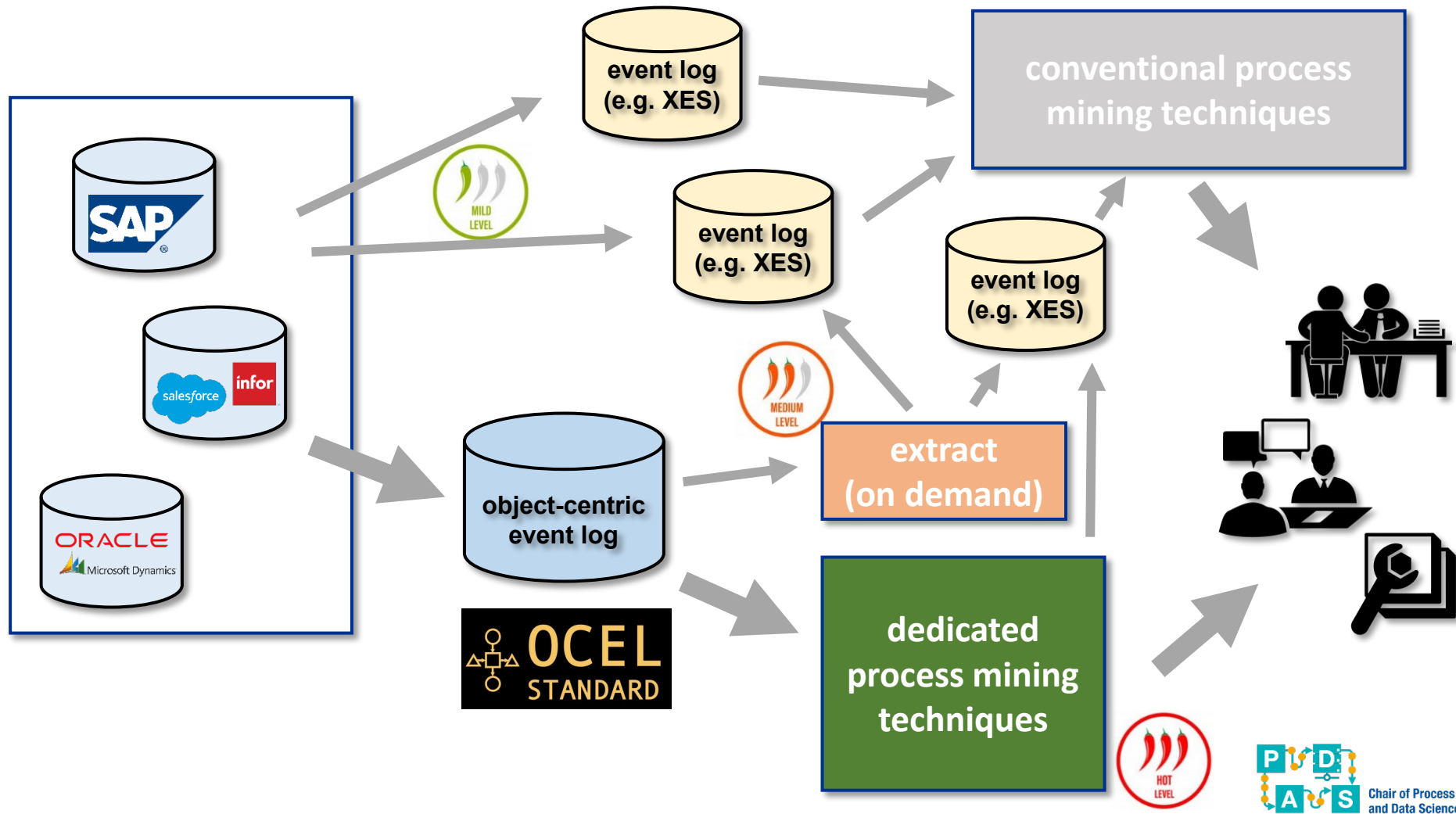
The purpose of the **OCEL** standard is to provide a general standard to interchange object-centric event data with multiple case notions. We set the following goals for the standard:

- **Interoperability:** with the provision of the OCEL standard and JSON/XML serializations of OCEL, we want to support a widespread collection of languages and systems.
- **Generalization:** the standard supports the storage of events, objects, and their attributes. Furthermore, the standard can be extended.
- **Provision of a collection of examples:** example logs, extracted from information systems supporting some widespread business processes, are provided for the OCEL standard.
- **Tool/Library Support:** to support the implementation of OCEL in custom applications, tool/library support shall be provided.



```
<events>
  <event>
    <string key="id" value="e1"/>
    <string key="activity" value="place_order"/>
    <date key="timestamp" value="2020-07-09T08:20:01.527+01:00"/>
    <list key="omap">
      <string key="object-id" value="i1"/>
      <string key="object-id" value="o1"/>
      <string key="object-id" value="i2"/>
    </list>
    <list key="vmap">
      <string key="resource" value="Alessandro"/>
      <float key="prepaid-amount" value="200.0"/>
    </list>
  </event>
  <event>
    <string key="id" value="e2"/>
    <string key="activity" value="check_availability"/>
    <date key="timestamp" value="2020-07-09T08:21:01.527+01:00"/>
  </event>
</events>
```

<http://ocel-standard.org/>



Conclusion

Lessons Learned



Lessons Learned

- Keep it **simple**, but not too simple.
- Learn to crawl before you can walk (i.e., **use XES first**).
- Be clear about the **core concepts**, e.g., is an event atomic or not (if so, you need activity instances).
- Need to be able to **relate events** in a **flexible** manner.
- Need to be able to **relate events and objects** (see ▲).
- We need **domain specific ontologies** instead of standard extensions, e.g., the P2P and O2C ontology.
- **Concepts first**: Multiple exchange formats.

