Database Process Mining: A Log/Data Meta Model

Eduardo González López de Murillas
Hajo Reijers
Wil van der Aalst

October 29th 2015
We live in a 3D world
We tend to observe reality in 2D
We see things in a different way
Sometimes unusual
But that is the way we work best
However, some perspectives do not work every time
It is good we can go back to the source.
To look at things differently
Our research field looks at systems with volume
From a flat point of view

<table>
<thead>
<tr>
<th>71</th>
<th>451</th>
<th>368</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>164</td>
<td>94</td>
</tr>
<tr>
<td>235</td>
<td>166</td>
<td>172</td>
</tr>
<tr>
<td>433</td>
<td>896</td>
<td>2.132</td>
</tr>
<tr>
<td>.870</td>
<td>2.845</td>
<td>1.001</td>
</tr>
<tr>
<td>2.427</td>
<td>1.133</td>
<td>1.308</td>
</tr>
<tr>
<td>2.424</td>
<td>2.697</td>
<td>1.710</td>
</tr>
<tr>
<td>1.692</td>
<td>1.844</td>
<td>1.725</td>
</tr>
<tr>
<td>1.199</td>
<td>1.903</td>
<td>1.442</td>
</tr>
<tr>
<td>2.032</td>
<td>1.198</td>
<td>2.453</td>
</tr>
<tr>
<td>2.390</td>
<td>3.850</td>
<td>2.175</td>
</tr>
<tr>
<td>1.920</td>
<td>1.748</td>
<td>2.387</td>
</tr>
<tr>
<td>3.928</td>
<td>3.176</td>
<td>2.514</td>
</tr>
<tr>
<td>1.287</td>
<td>1.272</td>
<td>2.303</td>
</tr>
<tr>
<td>2.110</td>
<td>1.928</td>
<td>1.928</td>
</tr>
<tr>
<td>3.292</td>
<td>3.393</td>
<td>2.960</td>
</tr>
<tr>
<td>1.272</td>
<td>1.928</td>
<td>1.851</td>
</tr>
<tr>
<td>110</td>
<td>393</td>
<td>222</td>
</tr>
<tr>
<td>249</td>
<td>277</td>
<td>175</td>
</tr>
<tr>
<td>304</td>
<td>2402</td>
<td>1.988</td>
</tr>
<tr>
<td>309</td>
<td>2402</td>
<td>1.988</td>
</tr>
<tr>
<td>450</td>
<td>451</td>
<td>367</td>
</tr>
<tr>
<td>182</td>
<td>139</td>
<td>144</td>
</tr>
<tr>
<td>232</td>
<td>377</td>
<td>431</td>
</tr>
<tr>
<td>430</td>
<td>451</td>
<td>367</td>
</tr>
<tr>
<td>182</td>
<td>139</td>
<td>144</td>
</tr>
<tr>
<td>340</td>
<td>301</td>
<td>336</td>
</tr>
<tr>
<td>232</td>
<td>377</td>
<td>431</td>
</tr>
<tr>
<td>430</td>
<td>451</td>
<td>367</td>
</tr>
</tbody>
</table>
We lose information in the process.
Our Logs are A point of view on data

A flat version of something more complex
Can we improve our point of view?

- More complete
- Less ambiguous
- More connected
In addition to what we already know
We propose a more complete model
How things happen

Process

How things are stored and linked

Data Schema
Medium abstraction layer

Each case for the process

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Size</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Shoe</td>
<td>42</td>
<td>35.7</td>
</tr>
<tr>
<td>2</td>
<td>T-shirt</td>
<td>M</td>
<td>15.8</td>
</tr>
<tr>
<td>3</td>
<td>Pants</td>
<td>31</td>
<td>41.5</td>
</tr>
</tbody>
</table>
Low abstraction layer

Most basic element on this side

Each of the historical states objects went through

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Size</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Shoe</td>
<td>42</td>
<td>35.7</td>
</tr>
<tr>
<td>1</td>
<td>Shoe</td>
<td>42</td>
<td>30.0</td>
</tr>
<tr>
<td>2</td>
<td>T-shirt</td>
<td>M</td>
<td>15.8</td>
</tr>
<tr>
<td>2</td>
<td>T-shirt</td>
<td>L</td>
<td>15.8</td>
</tr>
<tr>
<td>2</td>
<td>T-shirt</td>
<td>M</td>
<td>10.5</td>
</tr>
<tr>
<td>3</td>
<td>Pants</td>
<td>31</td>
<td>41.5</td>
</tr>
<tr>
<td>3</td>
<td>Pants</td>
<td>32</td>
<td>41.0</td>
</tr>
</tbody>
</table>
Low abstraction layer

Abstraction and detail
Meta Model in detail
Meta Model in detail

[Diagram of Meta Model with detailed elements labeled, including objects, events, data schema, and processes.]

AIS - 2015-10-29
Meta Model in detail
Ideal scenario

- The structure is filled with connected data

- Then we can work with it and make queries
  - SQL
  - Domain Specific Language (POQL)
What does this Meta Model provide?

• Common structure to focus on techniques to analyse data and ask questions:
  • Versions of an Object
  • Relations between Versions
  • Attributes of an event
  • Events of a case
  • Objects of a class
  • Events linked to a version
More complex questions

- Objects that (at some point) had a value for an attribute
- Objects linked to other objects
- Find objects affected by events of a specific task.
- Which case and event is involved in the modification of a field in an object?:
  - “Which activity in the process triggered the modification of the price of a product?”
Simple example

Customers for which the status value has changed from Silver to Gold membership, and are linked to an office located in Madrid.
Classical approach

- Customers for which the status value has changed from Silver to Gold membership, and are linked to an office located in Madrid:

1. Filter all the offices located in Madrid.
2. Filter all the customers linked to such offices.
3. Gather all their events.
4. Check which of them had a Silver membership.
5. Find out which ones promoted to a Gold membership.

But remember: all you have are flat events.
Only with an event log: tedious task

- We loose relation between Customer and Office
- Necessary to identify Customers
  - Events with certain attributes?
  - What is the ID?
- To identify Offices
- And to find the link
  - What is the attribute holding the relation?
- Handling n:m relations becomes complicated

Data Model / Domain knowledge becomes necessary
Using SQL on our Meta Model

SELECT OBJCust.id
FROM object as OBJCust,
    class as CCust,
    class as COff,
version as OBJVSilver,
version as OBJVGold,
version as OBJVMadrid,
attribute_value as ATVStatusSilver,
attribute_value as ATVStatusGold,
attribute_name as ATStatus,
attribute_value as ATVLocation,
attribute_name as ATLocation,
object as OBJOff,
class as COff,
version as OBJVOFF,
attribute_value as ATVLocation,
attribute_name as ATLocation,
relation as REL
WHERE OBJCust.class_id == CCust.id
AND CCust == ATStatus.class_id
AND ATStatus.name == "status"
AND ATVStatusGold.object_version_id == OBJVGOld.id
AND ATVStatusSilver.object_version_id == OBJVSilver.id
AND ATStatus.id == ATVStatusGold.attribute_name_id
AND ATStatus.id == ATVStatusSilver.attribute_name_id
AND ATVStatusGold.value == "Gold"
AND ATVStatusSilver.value == "Silver"
AND OBJVGOld.object_id == OBJCust.id
AND OBJVSilver.object_id == OBJCust.id
AND OBJVGOld.start_timestamp >= OBJVSilver.end_timestamp
AND COff.name == "OFFICE"
AND OBJOff.class_id == COff.id
AND OBJVOFF.object_id == OBJOff.id
AND ATLocation.name == "Location"
AND ATLocation.class_id == COff.id
AND ATVLocation.attribute_name_id == ATLocation.id
AND ATVLocation.object_version_id == OBJVOFF.id
AND (REL.source_object_version_id == OBJVOFF OR
    REL.target_object_version_id == OBJVOFF)
AND (REL.source_object_version_id == OBJVSilver OR
    REL.source_object_version_id == OBJVGOld)
AND (REL.target_object_version_id == OBJVSilver OR
    REL.target_object_version_id == OBJVGOld)
In POQL

objectsOf(
    versionsRelatedTo(
        versionsOf(
            allClasses where name == "OFFICE"
        ) where at.Location == "Madrid"
    ) where at.status changed from "Silver" to "Gold"
)
POQL: Process Oriented Query Language

- 10 first class types:
  - Activity
  - Activity Instance
  - Case
  - Event
  - Version
  - Relation
  - Object
  - Class
  - Attribute
  - Relationship
Some queries in POQL

Query 1:
versionsOf(
    allActivities where name contains "UPDATE"
) where start_timestamp => "1293836400000"

Query 2:
versionsOf(
    allObjects where not class_id == "3"
) where (not start_timestamp => "1296500000000" and not end_timestamp == "-1")

Query 3:
activitiesOf(
    eventsOf(
        allVersions where at.Decision_ID changed from "3" to "1"
    )
)
LIVE DEMO

JUST DO IT
All of this is possible...

- If we have the right data
What if some information is missing?

- Worse case: only events are known
Scenario 1

- **Worse case: only events are known**

  If events are sufficiently annotated: attributes

Detection of:
1. Classes
2. Lds
3. Relationships
What if some information is missing?

• Only events and Data schema are known

Trace building with Trace ID Patterns (BPM 2015)
What if some information is missing?

- Only events and instances are known
What if some information is missing?

- Only events and Data schema are known

Identifying values (from events) for the ID fields of each class (from Data Schema)
What if some information is missing?

- Only events, Data schema and objects are known

Identifying values for attributes of each class, grouping per object and ordering in time.
What if some information is missing?

- Only Data schema, objects and versions are known

One event per version (represents a change in an object)
We can reconstruct the whole structure

- Events
- Versions
- Instances
- Objects
- Process
- Data Schema
Conclusion

- Meta Model provides:
  - Structure
  - Context
  - A “less flat” view on the process

- Allows to focus on the analysis

- Makes easier to query and filter information

- Sets the basis for further studies

- When some information is missing, it can be inferred