Semantic Technologies for Digital Twins

Diego Calvanese

Free University of Bozen-Bolzano, Italy

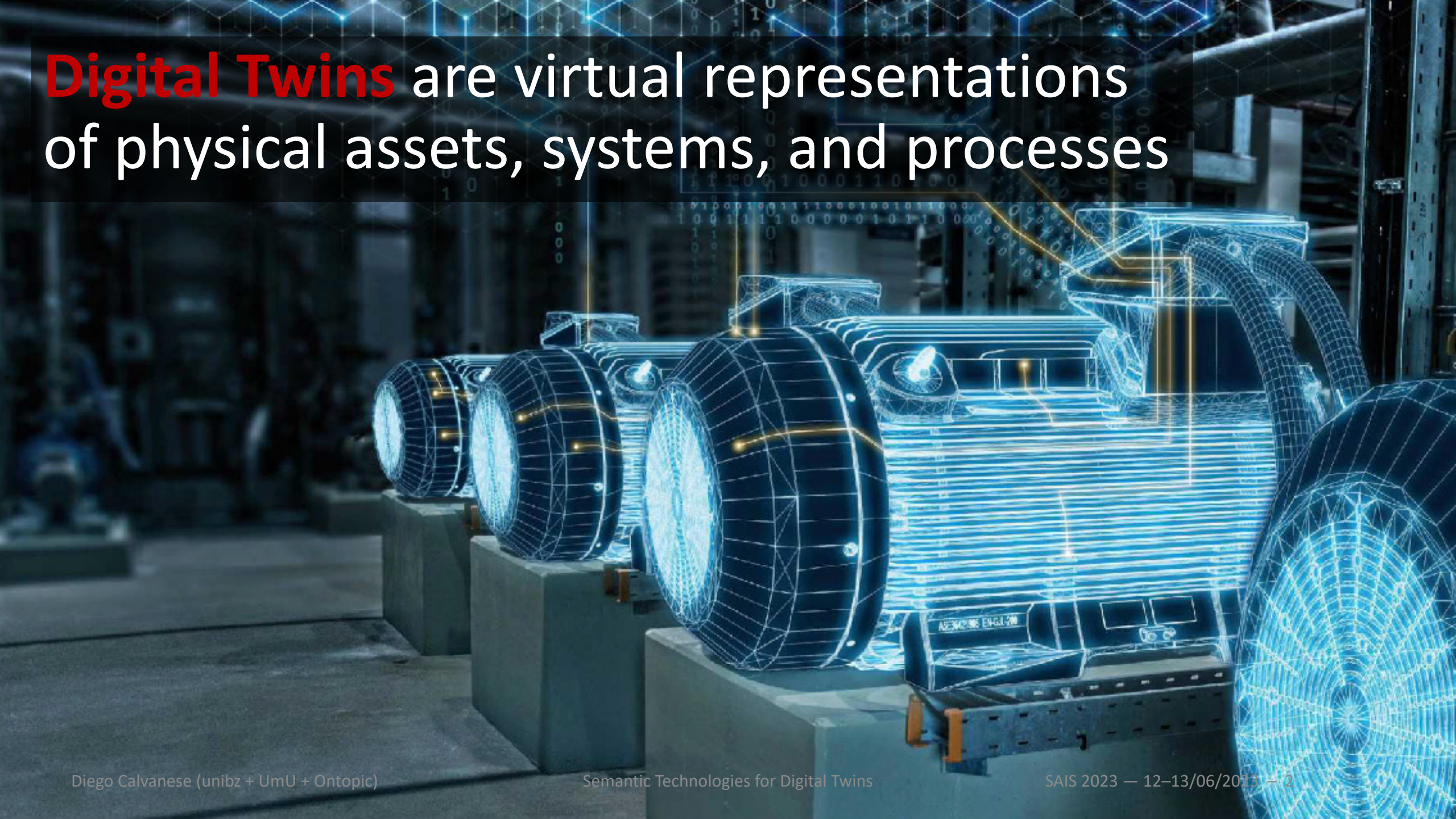
Umeå University, Sweden

Ontopic s.r.l., Italy

35th Swedish Artificial Intelligence Society Annual Workshop (SAIS 2023)
12–13 June 2023 – Karlskrona, Sweden

unibz

Digital Twins are virtual representations of physical assets, systems, and processes



Use of Digital Twins

Digital Twins are becoming increasingly popular in different domains, to optimize operations, minimize downtime, and enhance product quality.

To realize their full potential, Digital Twins:

- must become part of the broader enterprise ecosystem
- must **correctly model**
  - **static aspects**, i.e., objects, with their components and their relationships
  - **dynamic aspects**, i.e., how these objects evolve over time
- must leverage **data from multiple heterogeneous sources**

Constructing Digital Twins – Model-driven approach

The idealized reality of modelers and analysts
... as studied and planned by using different types of **models**.

Constructing Digital Twins – Data-driven approach

The actually experienced reality
... as wittnessed by information system **extracted data and logs**.

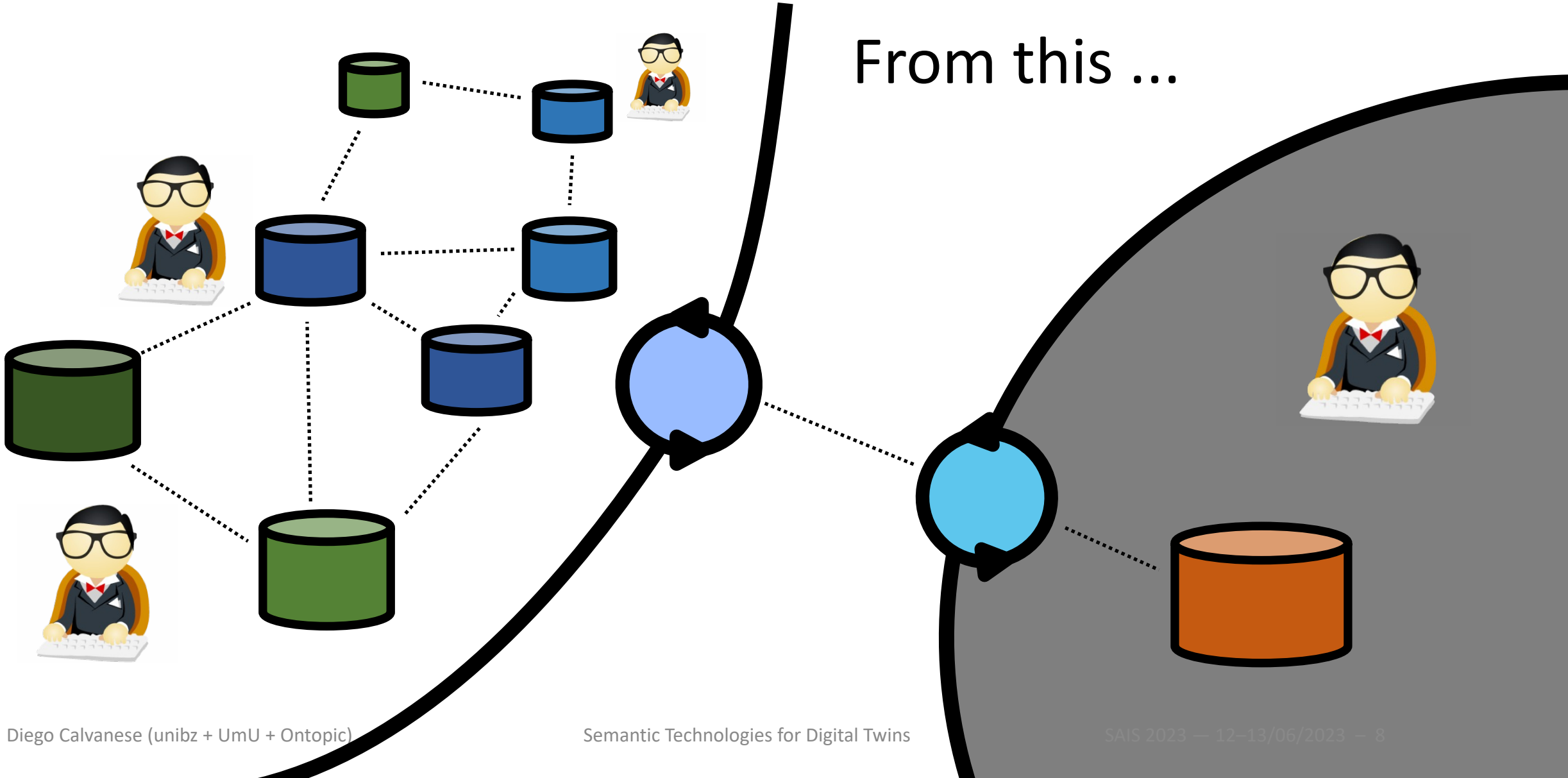
Constructing Digital Twins – The impact of Semantic Technologies

- Semantic Technologies provide a solid foundation for the model-driven approach
- They provide also significant support to the data-driven approach, by facilitating data access and preparation
- On the other hand, data-driven and ML techniques are needed to facilitate the setup of solutions based on Semantic-Technologies

Static aspects – Objects and their relationships

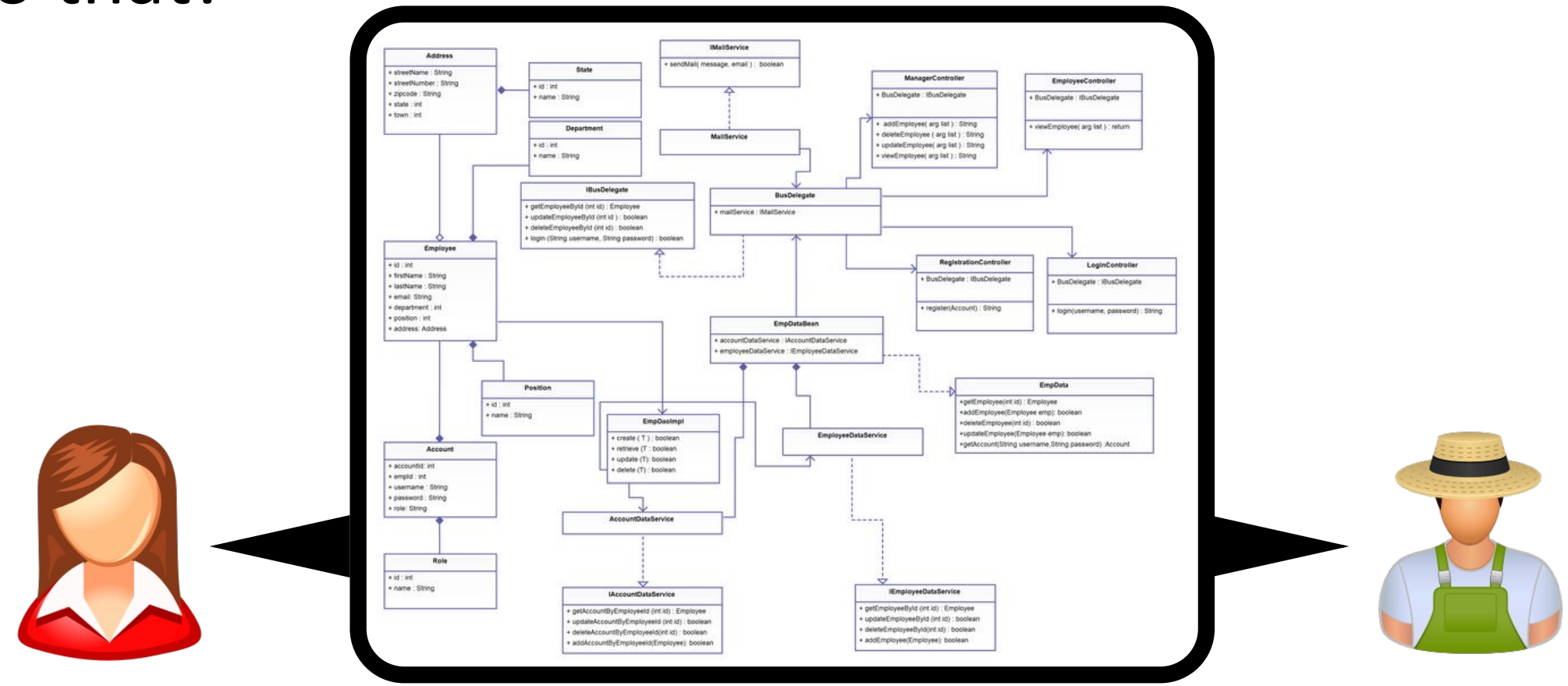
Why Semantic Technologies for Digital Twins

From this …



Why Semantic Technologies for Digital Twins

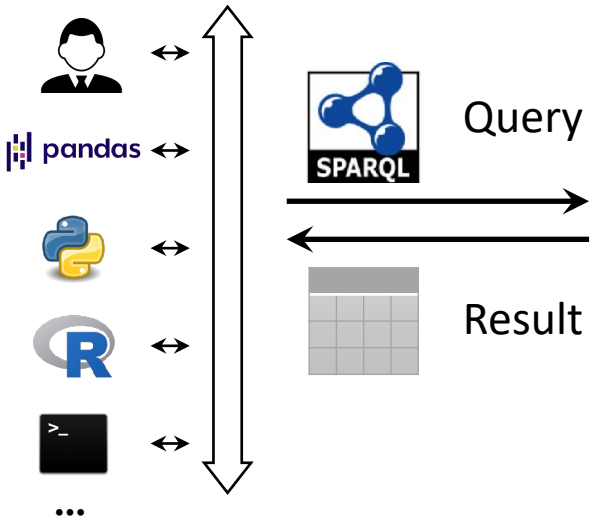
… to that!

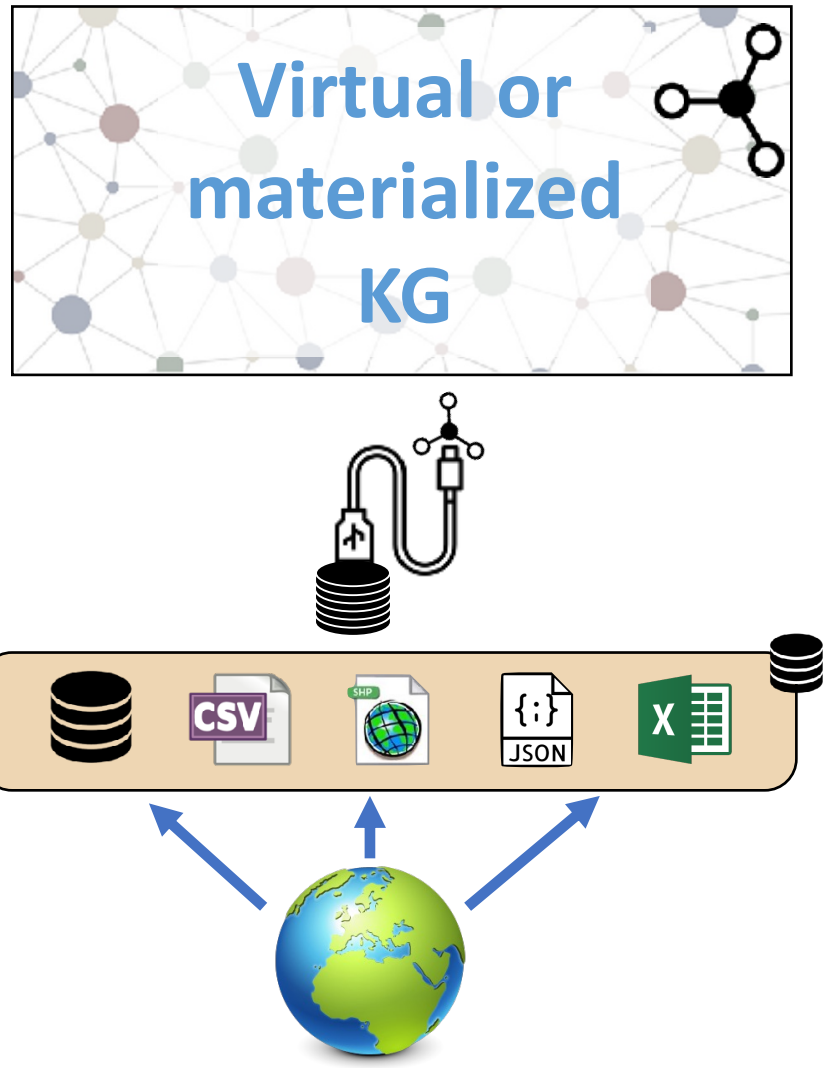


Semantic Technologies rely on Ontologies

- Provide convenient **abstraction mechanisms**
- Allow for high-level modeling of the **domain of interest**
- Are **logic-based**, hence come with a formal semantics
- **Automated reasoning tools** provide inference capabilities
- Rely on **standard languages** (W3C) – RDF, OWL2, R2RML, SPARQL
- Extensive **tool support** – parsing, visualization, editing, ...
- Provide a **bridge to the wider enterprise eco-system**

(Virtual) Knowledge Graphs as Digital Twins





Lightweight ontology

Mapping

Data sources

VKG framework is also known as Ontology-based Data Access / Integration (OBDA/I)

Languages for (V)KGs – Standardized by W3C

- **(V)KG**: expressed as a (virtual) **RDF graph**
- **Ontology**: expressed in **OWL 2 QL**, a lightweight profile of OWL 2
- **Data sources**: are **relational DBs**, possibly exposed through a data federation layer (e.g., Dremio, Denodo, Teeid, Apache Spark)
- **Mappings**: expressed in **R2RML**
  - SQL query over the data sources is mapped to class / property of the ontology
  - make use of IRI-templates to construct ontology objects from database values – overcomes impedance-mismatch between ontology and data source levels
- Language for querying the KG: **SPARQL**, the standard RDF query language

Example: Siemens Energy Services

- Monitor gas and steam turbines
- Collect data from 50 remote diagnostic centers around the world
- Centers linked to a common central DB
- Turbines are highly complex, with 5000–50000 sensors each
- Engineers compute KPIs and extract data from maintenance reports using ETL tools

Example: Digital Twins for gas turbines

- Structural components and functional units of gas turbines modeled through an ontology
- Digital Twin abstracts from individual sensors and specificity of each turbine
- Diagnostics programs are formulated on Digital Twin, and can be ported across devices
- Significantly simplifies monitoring and maintenance

Challenges for designing KG-based Digital Twins

1. Ontology and mappings might become very large and complex
2. Their design is labor-intensive and error prone, and requires substantial manual effort
3. Dealing with specific data types, e.g., geospatial data, raster data, ...

Possible solutions

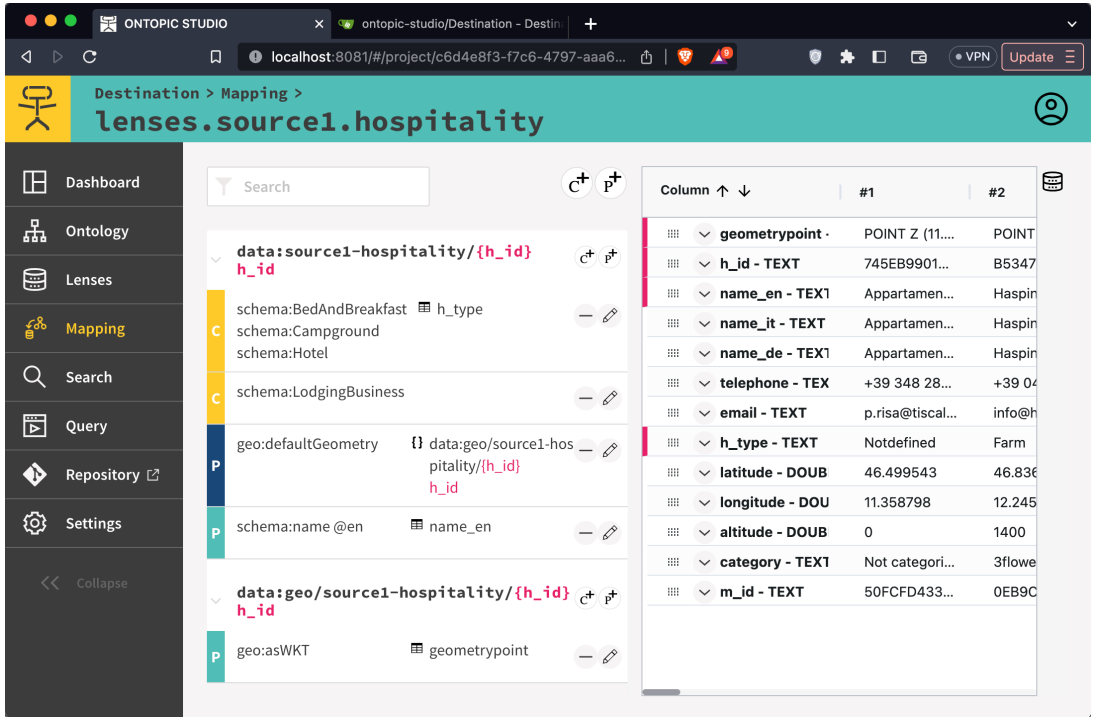
1. Rely on: design methodologies, tailored tools with user-friendly interfaces, abstraction
2. Automate ontology and mapping extraction from data sources
3. Develop tailored solutions that account for special data semantics

Ontology and mapping design

Ontology-design is **well-studied** with **established methodologies**, e.g., Ontoclean

Mapping design is **more critical**:

- Form of mappings heavily affects performance
- One can use **tailored mapping editors**, e.g., Ontopic Studio
- **Automate** (ontology- and) **mapping extraction** from available data sources



Mapping extraction based on mapping patterns

- Builds on well-established methodologies and patterns studied in data management and conceptual modeling
- We have defined a **catalog of KG mapping patterns** that consider:
  - the domain knowledge that is encoded in **ontology axioms**
  - the relational **DB schema** with its **constraints** (keys, foreign keys, ...)
  - the **conceptual schema** at the basis of the relational DB schema
  - **actual data** stored in the sources (when available)
- Patterns identified in schema and data allow for extracting mappings (and ontology axioms)

Two types of mapping patterns

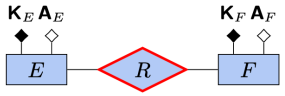
Schema-driven patterns

- Are shaped by the structure of the DB schema and its explicit constraints

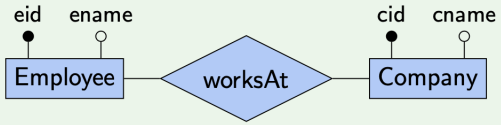
Data-driven patterns

- Consider constraints emerging from specific configurations of the data
- Each schema-driven pattern has a data-driven counterpart: constraints over the schema are not explicitly specified, but hold in the data.
- Also specific data-driven patterns (without schema-driven counterpart)

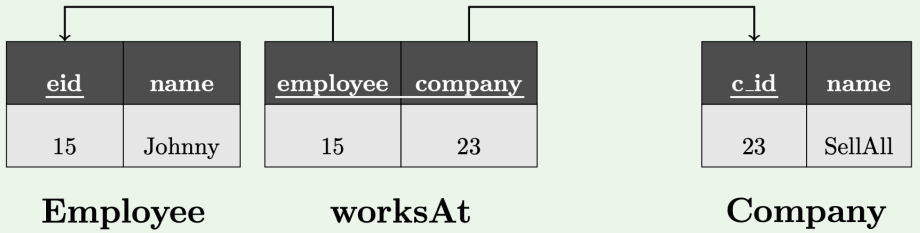
Example of schema-driven patterns

E-R DIAGRAM	DB SCHEMA	MAPPING	ONTOLOGY
Schema Entity (SE)			
	$T_E(\underline{\mathbf{K}}, \mathbf{A})$	s: T_E t: $C_E(\mathbf{iri}_E(\mathbf{K}))$, $\{d_A(\mathbf{iri}_E(\mathbf{K}), A)\}_{A \in \mathbf{K} \cup \mathbf{A}}$	$\{\exists d_A \sqsubseteq C_E\}_{A \in \mathbf{K} \cup \mathbf{A}}$
Schema Relationship (SR)			
	$T_E(\underline{\mathbf{K}_E}, \mathbf{A}_E)$ $T_F(\underline{\mathbf{K}_F}, \mathbf{A}_F)$ $T_R(\underline{\mathbf{K}_{RE}, \mathbf{K}_{RF}})$	s: T_R t: $p_R(\mathbf{iri}_E(\mathbf{K}_{RE}), \mathbf{iri}_F(\mathbf{K}_{RF}))$	$\exists p_R \sqsubseteq C_E$ $\exists p_R^- \sqsubseteq C_F$
In case of $(\_, 1)$ cardinality on role R_E (resp., R_F), the primary key for T_R is restricted to the attributes $\mathbf{K}_{RE}$ (resp., $\mathbf{K}_{RF}$).			

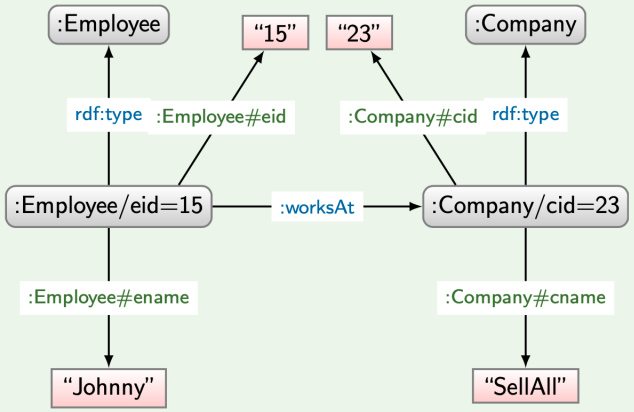
Conceptual



DB Schema



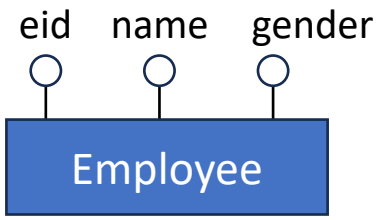
RDF (data only)



Example of data-driven pattern

Clustering Entity to Class (CE2C)

ER Diagram



$partition_D$(Employee,gender)

DB Schema

T_{Emp}(<u>eid</u>,name,gender)

$partition_D$(Employee,gender)

{ V_{EmpM}(eid,name,gender) = $\sigma_{gender='M'}$(Employee),

V_{EmpF}(eid,name,gender) = $\sigma_{gender='F'}$(Employee) }

Mapping

m1: $\sigma_{gender='M'}$(Employee) $\rightsquigarrow$ EmpM(e(eid))

m2: $\sigma_{gender='F'}$(Employee) $\rightsquigarrow$ EmpF(e(eid))

Ontology

EmpM $\sqsubseteq$ Emp

EmpF $\sqsubseteq$ Emp

eid	name	gender	
1	Johnny	M	Male
2	Elena	F	Female
3	Ann	F	
4	Paul	M	Male

Employee

Discovering data-driven patterns

- We rely on data-profiling techniques to discover exact and approximate data dependencies
- We use machine learning for
  - recognizing specific types of data
  - data clustering
  - recognizing correspondences between source and global schema elements (similar to what done in schema and ontology matching)
- We are currently exploring the use of Large Language Models for mapping extraction

Dynamic aspects – Events and processes

Processes leave digital footprints

Within organizations and industries: event data related to executions of processes are continuously generated and stored for

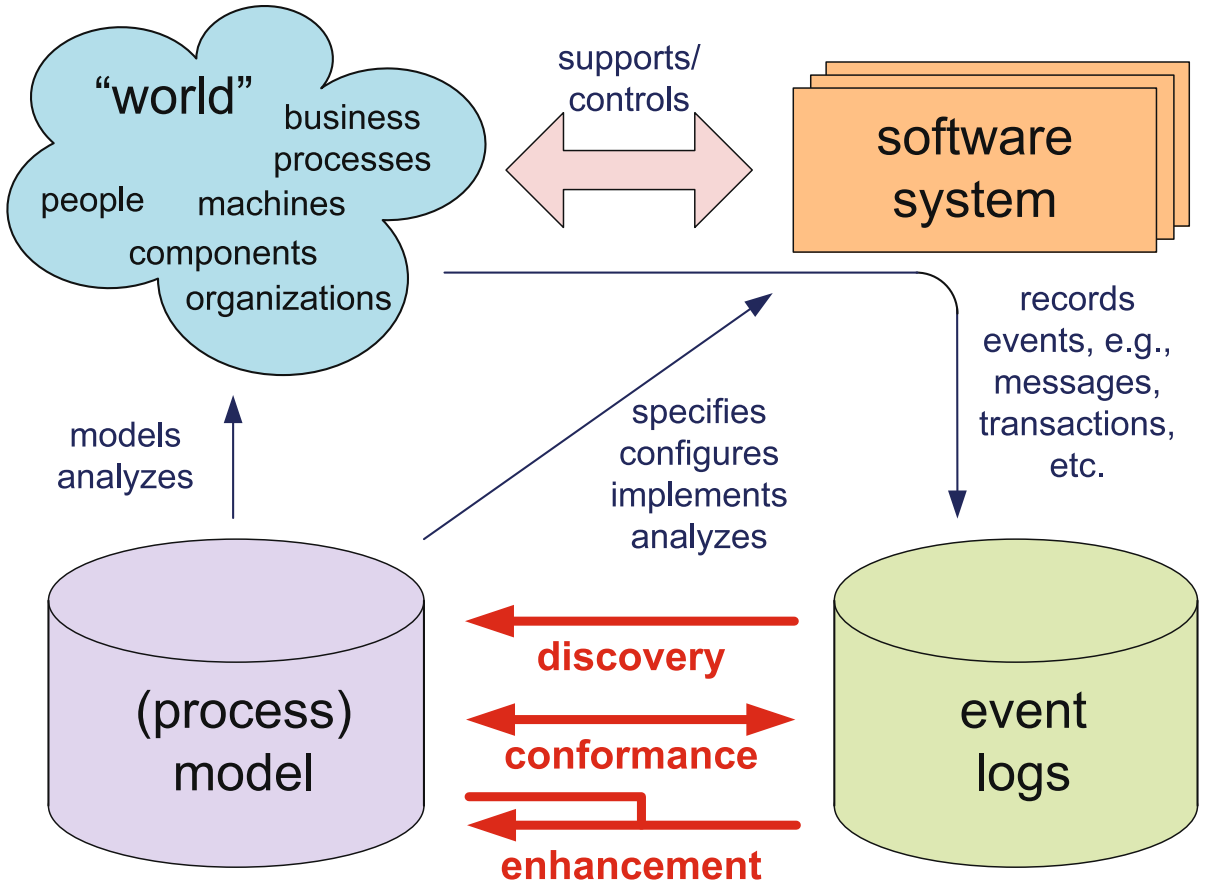
- needs of the organization/industry, e.g., for sensor data
- internal management
- calculation of process metrics / KPIs
- legal reasons (compliance, external audits)

In addition: internally and externally, more data are stored, e.g., mobile devices log data, vehicle data, social networks, ...

All such data contains valuable information for the organization/industry.

Process Mining: Process management based on facts

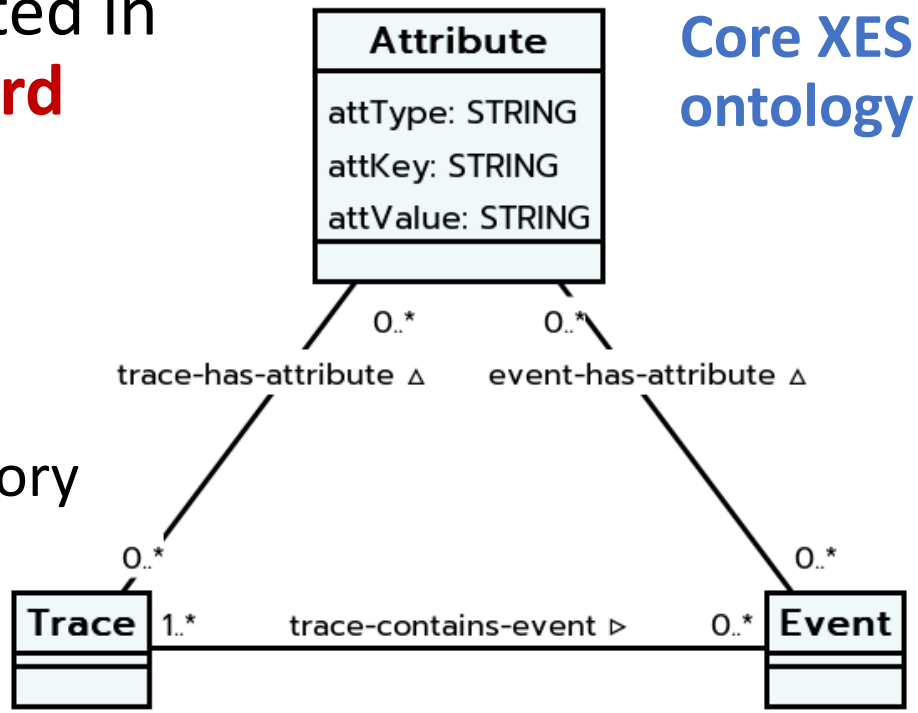
[see process mining manifesto]



Standard languages for representing event logs

Process mining tools rely on logs being represented in standard languages, notably the **IEEE XES standard** [www.xes-standard.org]

- Based on XML
- Minimal mandatory structure for a log:
  - **log** consists of **traces**, each representing a case history
  - trace consists of a **list of atomic events**
- Extensions to “decorate” log, trace, event with informative attributes (e.g., timestamps, task names, transactional lifecycle, resources, ...).



Core XES ontology

Logs of different quality – Where do they come from?

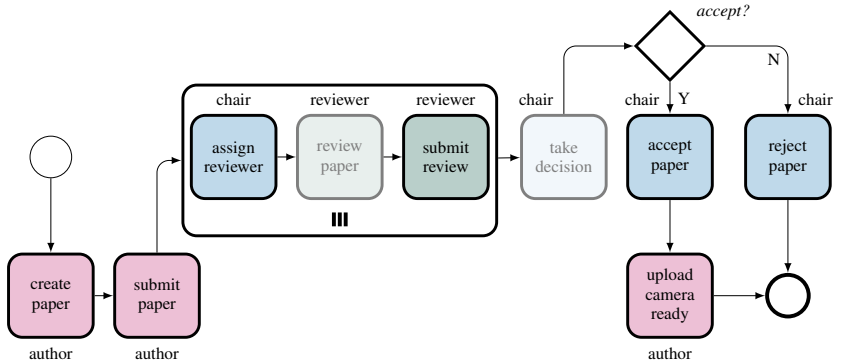
Level	Characterization	Examples
★★★★★ 5	Highest level: the event log is of excellent quality (i.e., trustworthy and complete) and events are well-defined. Events are recorded in an automatic, systematic, reliable, and safe manner. Privacy and security considerations are addressed adequately. Moreover, the events recorded (and all of their attributes) have clear semantics. This implies the existence of one or more ontologies. Events and their attributes point to this ontology.	Semantically annotated logs of BPM systems.
★★★★ 4	Events are recorded automatically and in a systematic and reliable manner, i.e., logs are trustworthy and complete. Unlike the systems operating at level ★★★, notions such as process instance (case) and activity are supported in an explicit manner.	Events logs of traditional BPM/ workflow systems.
★★★ 3	Events are recorded automatically, but no systematic approach is followed to record events. However, unlike logs at level ★★, there is some level of guarantee that the events recorded match reality (i.e., the event log is trustworthy but not necessarily complete). Consider, for example, the events recorded by an ERP system. Although events need to be extracted from a variety of tables, the information can be assumed to be correct (e.g., it is safe to assume that a payment recorded by the ERP actually exists and vice versa).	Tables in ERP systems, event logs of CRM systems, transaction logs of messaging systems, event logs of high-tech systems, etc.
★★ 2	Events are recorded automatically, i.e., as a by-product of some information system. Coverage varies, i.e., no systematic approach	Event logs of document and product management systems,

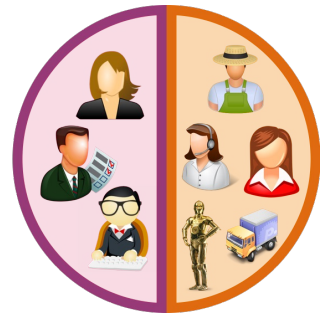
unibz

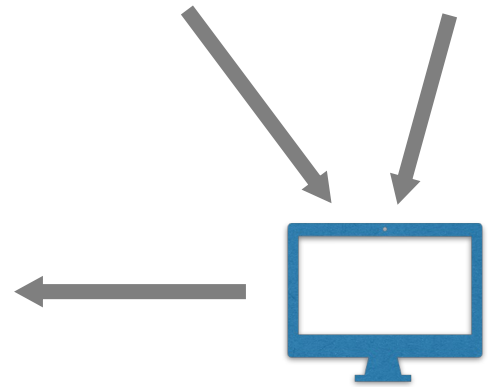
Situation 1: Explicit event logs – Levels 4–5

Organisations with **process-aware information systems**

- Support humans in process execution (task assignments, todo lists, ...)
- Explicitly log events, with info about:
  - timestamp
  - event type (start, end, reassign, …)
  - reference task
  - reference case
  - task instance id
  - responsible resource
  - additional attributes





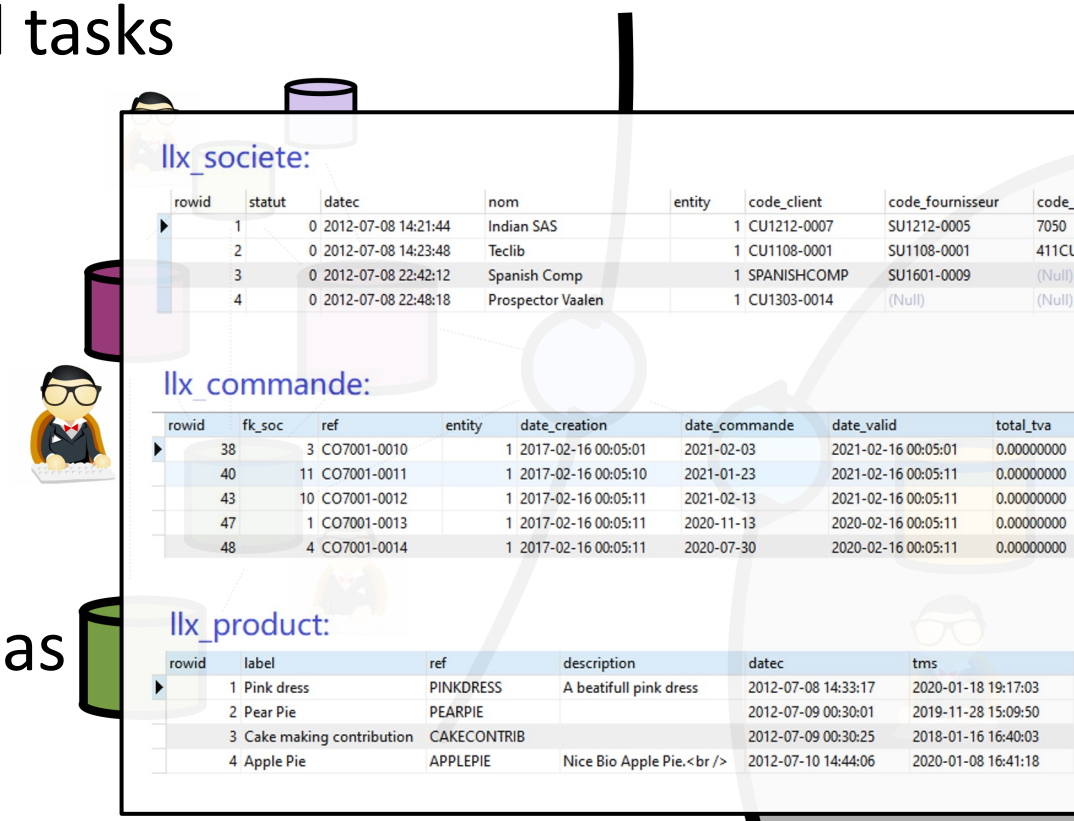


	Event Data				
Case ID	*ID*	*Timestamp*	*Activity*	*User*	...
1	35654423	30-12-2010:11.02	create paper	Pete	...
	35654424	31-12-2010:10.06	submit paper	Pete	...
	35654425	05-01-2011:15.12	assign review	Mike	...
	35654426	06-01-2011:11.18	submit review	Sara	...
	35654428	07-01-2011:14.24	accept paper	Mike	...
	35654429	06-01-2011:11.18	upload CR	Pete	...
2	35654483	30-12-2010:11.32	create paper	George	...
	35654485	30-12-2010:12.12	submit paper	John	...
	35654487	30-12-2010:14.16	assign review	Mike	...
	35654489	16-01-2011:10.30	submit review	Ellen	...
	35654490	18-01-2011:12.05	reject paper	Mike	...

Situation 2: Implicit event logs – Level 3

Organisations with **generic information systems**

- CRM, ERP systems to handle customers and tasks
- Legacy information systems
- Industrial systems
- Domain-specific systems



llx_societe:

rowid	statut	datec	nom	entity	code_client	code_fournisseur	code_
1	0	2012-07-08 14:21:44	Indian SAS	1	CU1212-0007	SU1212-0005	7050
2	0	2012-07-08 14:23:48	Teclib	1	CU1108-0001	SU1108-0001	411CU
3	0	2012-07-08 22:42:12	Spanish Comp	1	SPANISHCOMP	SU1601-0009	(Null)
4	0	2012-07-08 22:48:18	Prospector Vaalen	1	CU1303-0014	(Null)	(Null)

llx_commande:

rowid	fk_soc	ref	entity	date_creation	date_commande	date_valid	total_tva
38	3	CO7001-0010	1	2017-02-16 00:05:01	2021-02-03	2021-02-16 00:05:01	0.00000000
40	11	CO7001-0011	1	2017-02-16 00:05:10	2021-01-23	2021-02-16 00:05:11	0.00000000
43	10	CO7001-0012	1	2017-02-16 00:05:11	2021-02-13	2021-02-16 00:05:11	0.00000000
47	1	CO7001-0013	1	2017-02-16 00:05:11	2020-11-13	2020-02-16 00:05:11	0.00000000
48	4	CO7001-0014	1	2017-02-16 00:05:11	2020-07-30	2020-02-16 00:05:11	0.00000000

llx_product:

rowid	label	ref	description	datec	tms
1	Pink dress	PINKDRESS	A beatifull pink dress	2012-07-08 14:33:17	2020-01-18 19:17:03
2	Pear Pie	PEARPIE		2012-07-09 00:30:01	2019-11-28 15:09:50
3	Cake making contribution	CAKECONTRIB		2012-07-09 00:30:25	2018-01-16 16:40:03
4	Apple Pie	APPLEPIE	Nice Bio Apple Pie. 	2012-07-10 14:44:06	2020-01-08 16:41:18

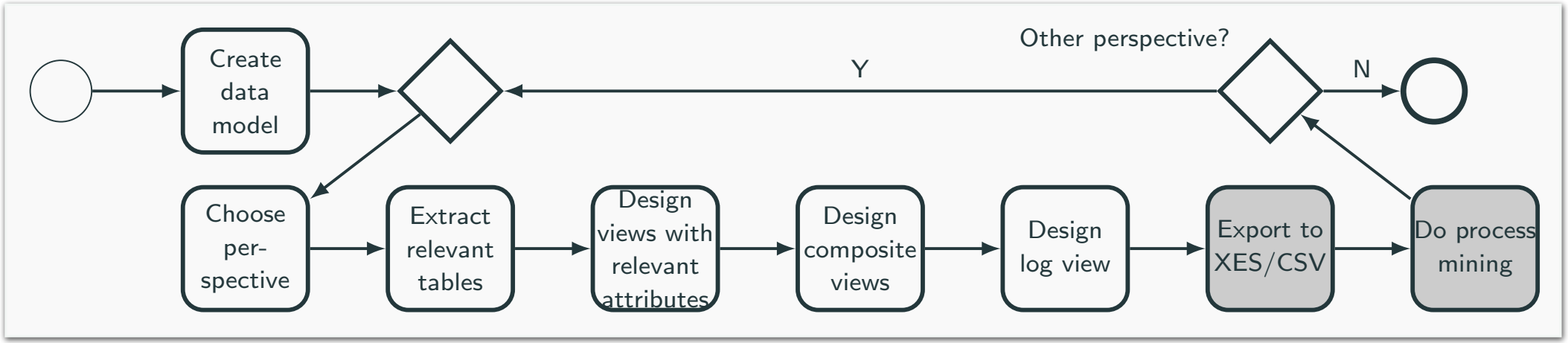
Data are stored with different formats and according to different domain-specific schemas

- No explicit events
- Data scattered in several data sources

From event logs to standard logs (in XES)

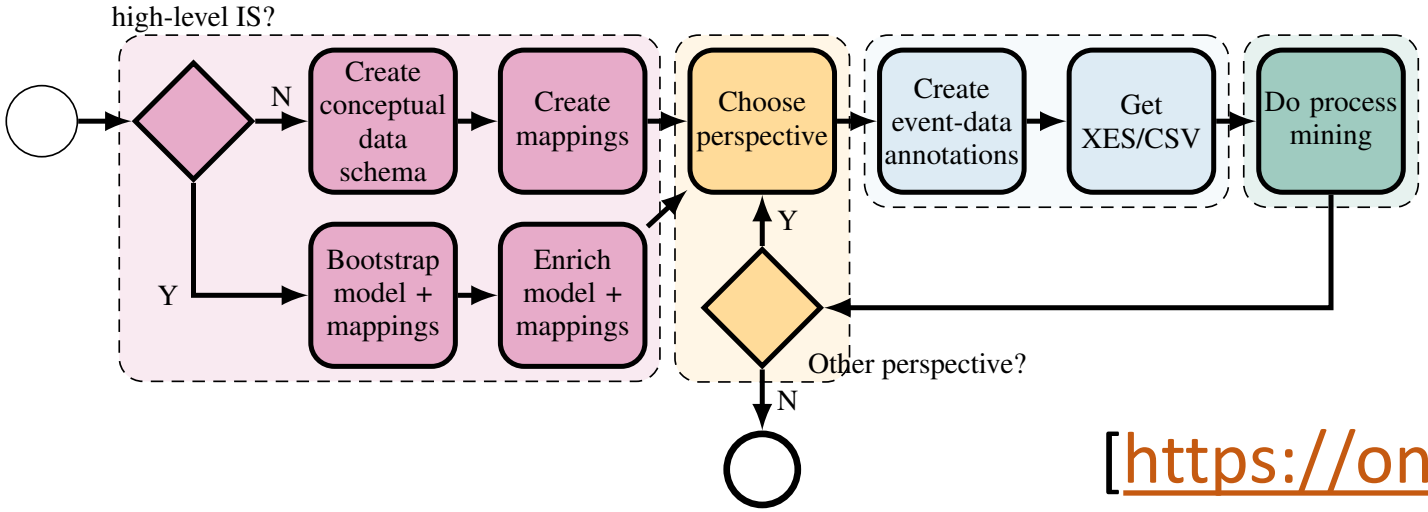
- Levels 4–5: straightforward syntactic manipulation
- Level 3: much more difficult, due to
  - multiple data sources
  - interpretation of data
  - lack of explicit information about cases and events

Traditional extraction from legacy data



- **Manual construction** of views and ETL procedures to fetch the data.
- Done by **IT experts**, not by knowledge workers (domain experts).
- Crucial issues:
  - correctness: who knows? Process mining is **dangerous** if applied on wrong data!
  - maintenance, evolution, change of perspective are hard …
    But process mining should be **highly interactive**.

The OnProm approach and toolchain



[https://onprom.inf.unibz.it]

OnProm relies on **Semantic Technologies** to:

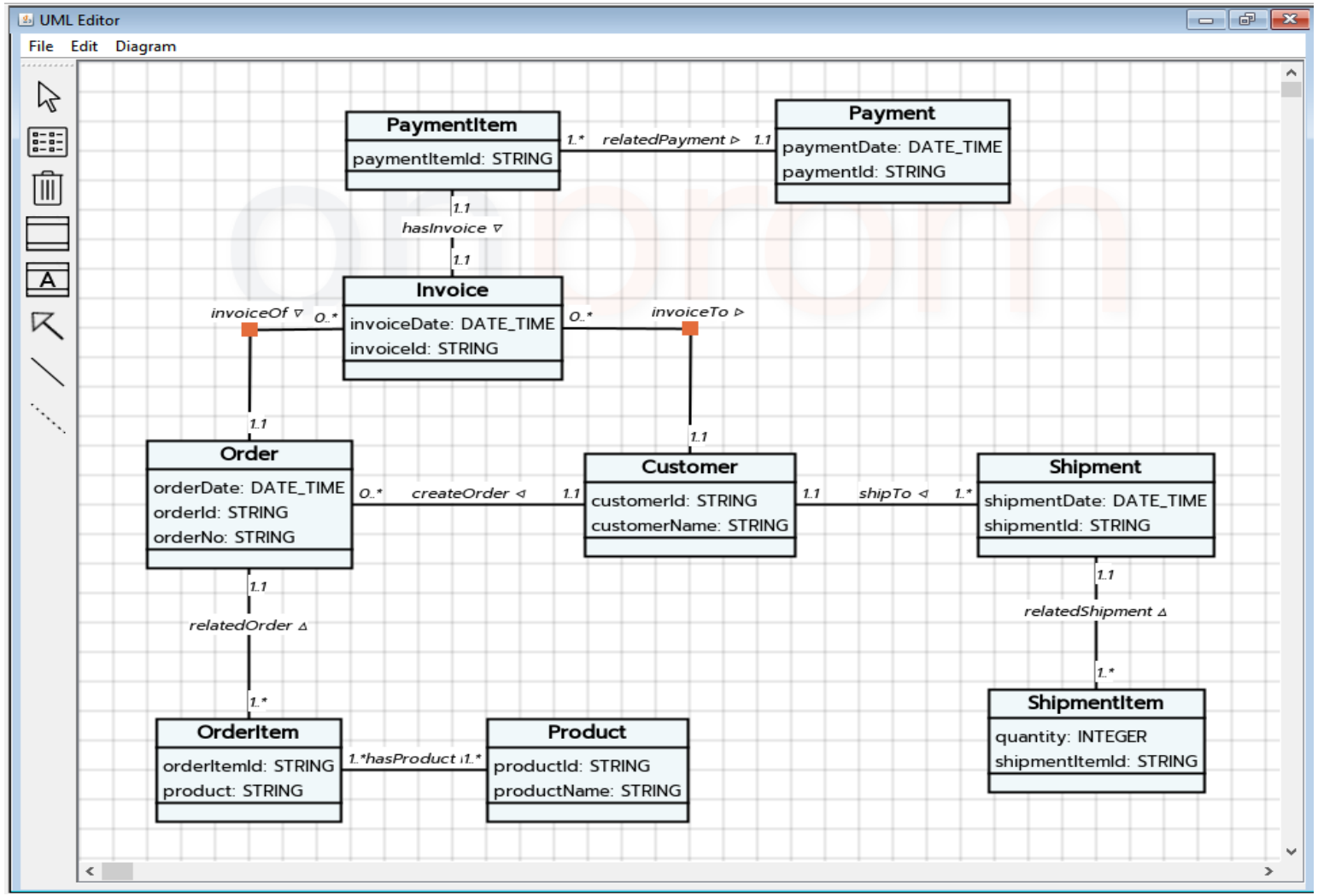
1. **Understand** the data.
2. **Access** the data using the **vocabulary of the domain ontology**.
3. Express the **perspective for process mining** using the **domain vocabulary**.
4. **Automatically extract** standard event logs.

The **OnProm** approach is **fully implemented in an integrated toolchain**.

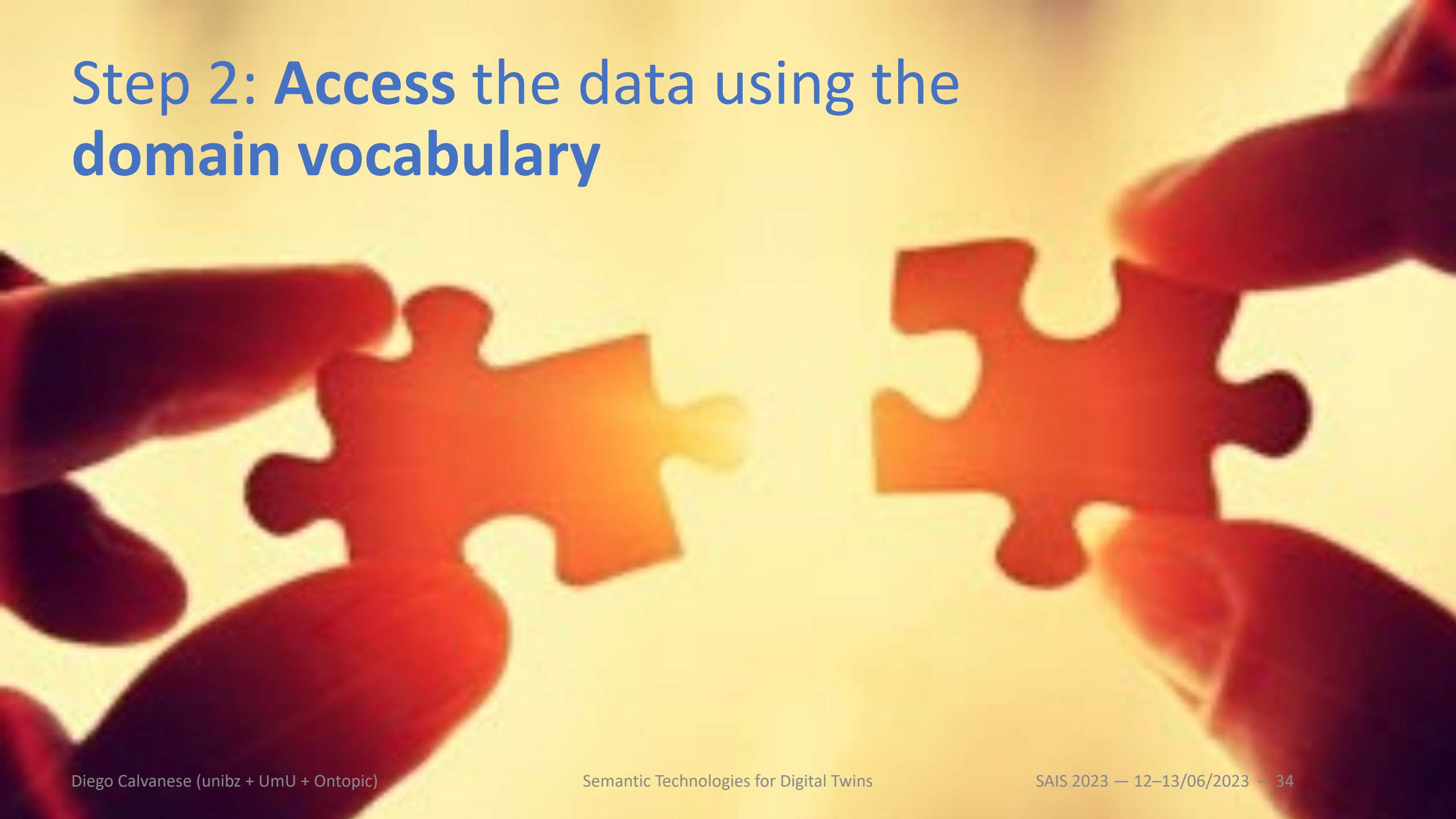
Step 1: **Understand** the data

Rely on a **domain model / ontology**

Conceptual schema / ontology formulated as **UML class diagram** and encoded in **OWL 2 QL**.

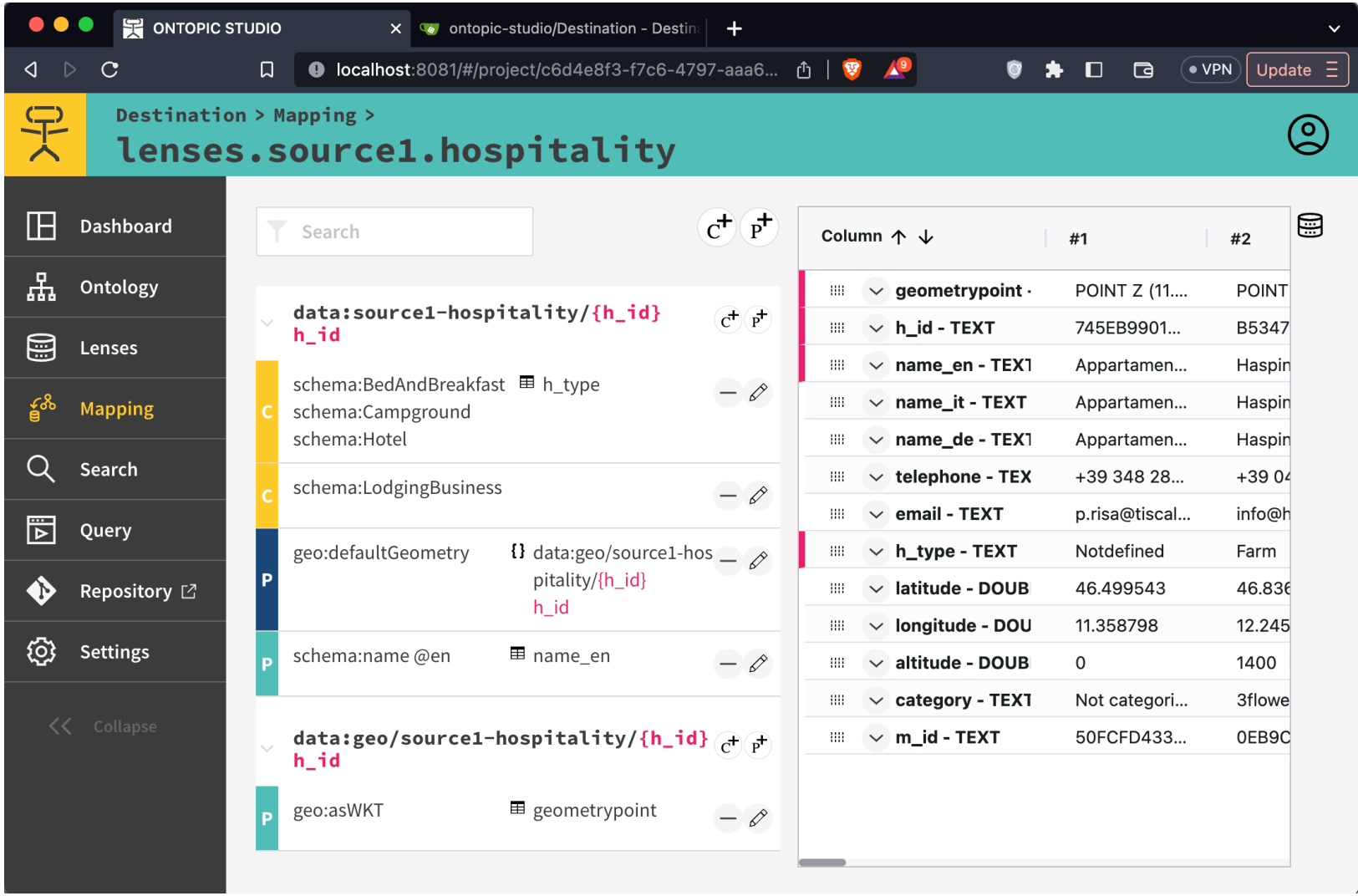


Step 2: **Access** the data using the **domain vocabulary**



We rely on **(V)KGs for data access**

- We connect the domain ontology to the underlying data source via **KG mappings**.
- To design the mappings, we rely on an external mapping editor, e.g., Ontopic Studio.



Step 3: Find the **event data**

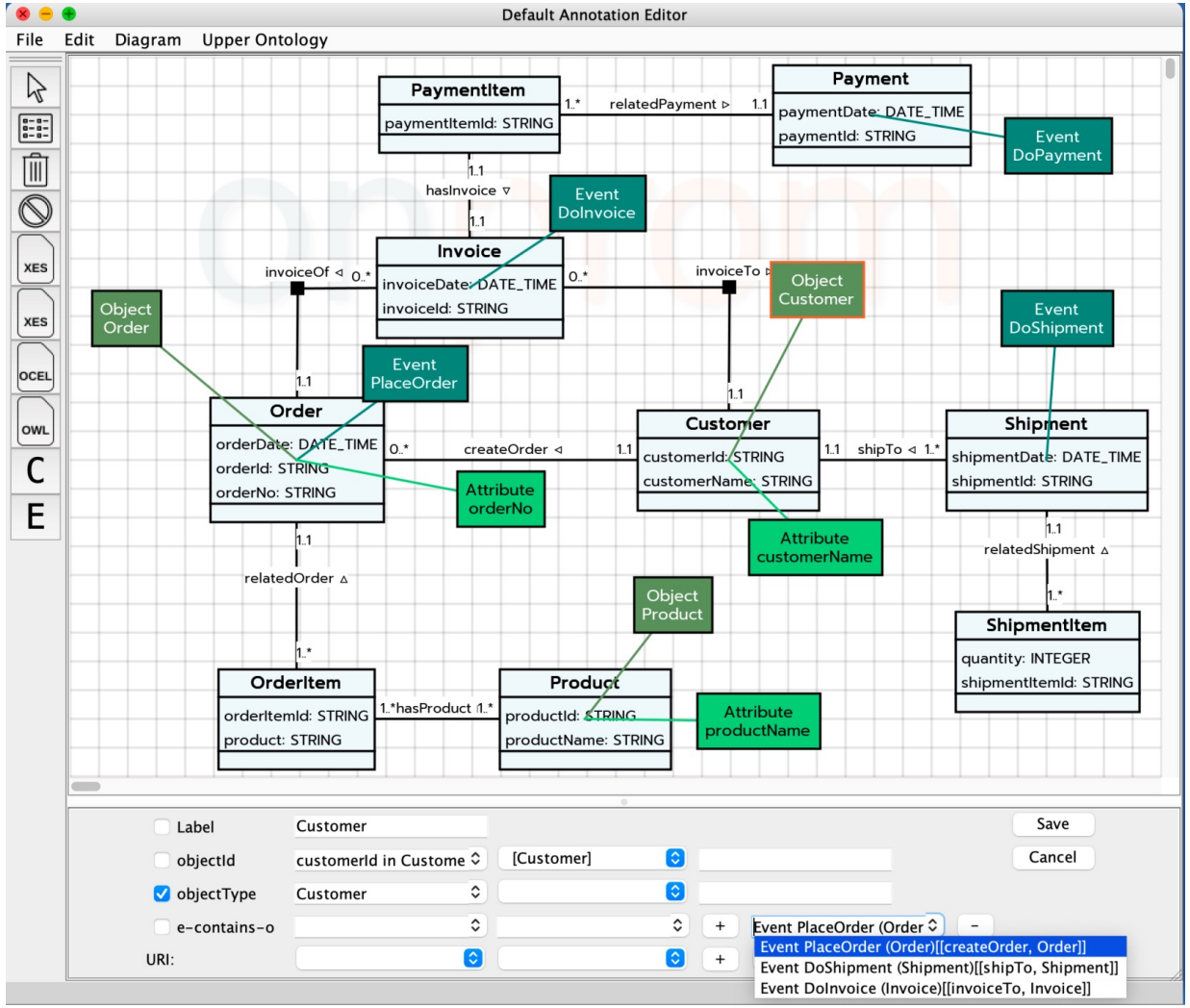
Annotating the conceptual data schema

Fix perspective: declare the **case object**:

- Find and annotate the class(es) whose instances are considered as case objects
- Express additional filters

Find the **events** (looking for timestamps):

- Find and annotate the classes whose instances refer to events
- Declare how they are connected to corresponding case objects
  —> navigation in the UML class diagram
- Declare how they are (in)directly related to event attributes (timestamp, task name, optionally event type and resource)
  —> navigation in the UML class diagram



Step 4: **Automatically extract** standard event logs

Formalizing and rewriting annotations

Annotations are nothing else than **SPARQL queries**
over the conceptual data schema

They can be **automatically rewritten into SQL queries**
over the legacy data making use of a VKG query rewriting engine

We automatically get a standard **VKG mapping**
from the XES ontology to the legacy data.

https://ontop-vkg.org/

- State-of-the-art (V)KG system
- Addresses the key challenges in query answering of scalability and performance
- Compliant with all relevant Semantic Web standards:

  RDF, RDFS, OWL 2 QL, R2RML, SPARQL, and GeoSPARQL
- Supports all major relational DBMSs and federation tools:

  Oracle, DB2, MS SQL Server, Postgres, MySQL, Teiid, Dremio, Denodo, etc.
- Open-source and released under Apache 2 licence

Querying the “virtual log”

SPARQL queries over the XES event schema are answered over legacy data using the (V)KG query answering engine.

Example: Get empty and nonempty traces; for nonempty traces, also fetch all their events.

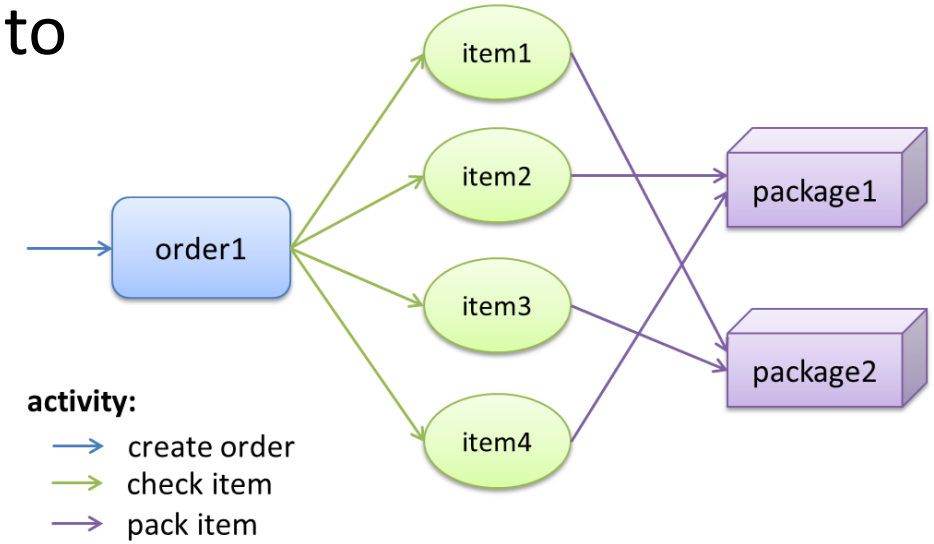
```
SELECT DISTINCT ?trace ?event
WHERE {
  ?trace a :Trace .
  OPTIONAL {
    ?trace :trace-contains-event ?event .
    ?event :event-has-attribute ?timestamp .
      ?timestamp :attKey “time:timestamp”^^xsd:string .
    ?event :event-has-attribute ?name .
      ?name :attKey “concept:name”^^sxd:string .
  }
}
```

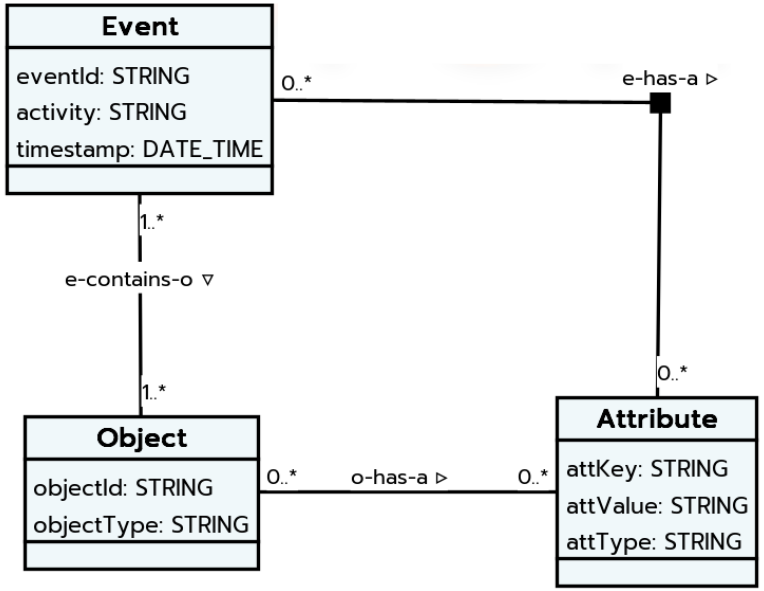
Answers are serialised into a fully compliant XES log!

From XES to **OCEL – Object-centric Event Logs**

Recently, the new more flexible OCEL standard for event logs has been proposed.

- An event can be associated to multiple objects (and not only to a single case, as in XES).
- An object is related to multiple events.



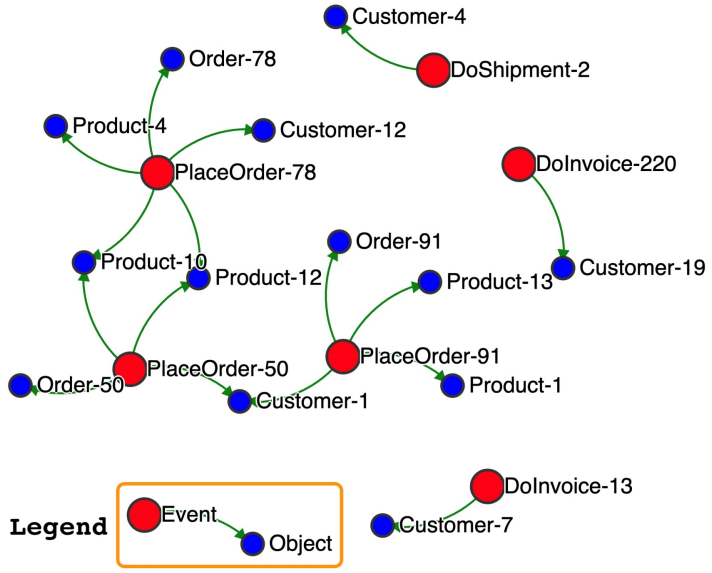


Adapting OnProm to OCEL

- We have extended the OnProm toolchain so as to support OCEL.
- The mechanism for the OCEL extraction is analogous to the one for XES, and relies on compiling the domain ontology and mappings together with the OCEL annotations into a new (V)KG specification.
- OCEL is actually a better fit than XES for the KG approach, which already exposes the data as a (virtual) knowledge graph.
- The adaptation to OCEL required a complete reimplementation of the core transformation components.
- The toolchain is now fully modular, and can be easily adapted to variations or extensions of the OCEL standard.

OCEL graph serialization

- OnProm extracts from the underlying data sources an internal representation of the OCEL graph in terms of Java objects.
- The OCEL graph can then be serialized in XML according to the OCEL standard.




```
    <event>
        <string key="id" value="PlaceOrder-50"/>
        <string key="activity" value="PlaceOrder"/>
        <date key="timestamp" value="2017-02-15T23:51:23"/>
        <list key="omap">
            <string key="object-id" value="Customer-1"/>
            <string key="object-id" value="Order-50"/>
            <string key="object-id" value="Product-10"/>
            <string key="object-id" value="Product-12"/>
            </list>
        <list key="vmap">
            <string key="orderNo" value="CO7001-0006"/>
        </list>
    </event>
</events>
<objects>
```

Challenges wrt the dynamic component of Digital Twins

- So far I just talked about the problem of event-log extraction
- How to correctly **represent and reason** about system dynamics?
  - Well-studied problem in Knowledge Representation
    reasoning about actions, temporal logics, planning, data-aware processes, ..
  - The spectrum of different options to choose from is much larger
  - Combination of static and dynamic aspects is notoriously difficult wrt reasoning – undecidability
- Proper **query languages** are still missing – temporal patterns, temporal granularity, streaming data, windowing, ...
- How to deal **efficiently** with data dynamics? – streaming data, raster data, ...

Conclusions

- Developing Digital Twins in complex domains requires capturing the static aspects of physical devices and their dynamics
- Semantic Technologies provide convenient abstraction mechanisms for setting up Digital Twins
  - (V)KGs can be used as (virtual) Digital Twins supported by underlying data sources
  - (V)KGs for data access can be used to prepare dynamically generated data for Process Mining, which in turn allows one to refine / improve Digital Twins
- A key challenge in using (V)KGs lies in setting up their components
- We are currently exploring novel approches based on machine learning and on exploiting Large Language Models for this task

Thank you!