Introduction to ODPs and first pattern examples

Pascal Hitzler
Data Semantics Laboratory (DaSe Lab)
Data Science and Security Cluster (DSSC)
Wright State University
http://www.pascal-hitzler.de
See IOS Press booth


Supplementary material for the chess example at [http://dase.cs.wright.edu/content/pattern-driven-linked-data-publishing-primer](http://dase.cs.wright.edu/content/pattern-driven-linked-data-publishing-primer)
This Tutorial (all parts)

- Pascal Hitzler (60 mins): Introduction and first examples
- Monika Solanki (30 mins): Example “modeling vaccine traceability”

coffee

- Pascal Hitzler (60 mins): Example “GeoLink Modular Ontology”
- Agnieszka Lawrynowicz (30 mins): Example “Reporting Event ODP”

lunch

- Karl Hammar with all others (3h): Hands-on, the WebProtege XDP plug-in
What this is about

• A tutorial about ontology modeling best practices.

• Coming from the “Ontology Design Patterns” community.

• Recommended by us for all types of ontology modeling, including as graph schema for linked data and knowledge graphs.

• We are approaching a point where our experiences can consolidate into crisp recommendations, but we’re not quite there yet. I.e. there’s still work (and research) to be done.

• Join us if you’re interested:
  Google Group called “Ontology Design Patterns”
Very first example

Generic AgentRole pattern

![Diagram of Generic AgentRole pattern]

Generic NameStub pattern

![Diagram of Generic NameStub pattern]

Joined:

![Joined Diagram]

May 2017 – ESWC 2017, Portoroz, Slovenia – Pascal Hitzler
Very first example

Joined AgentRole and NameStub patterns:

Used as a template for a concrete modeling problem:
Very first example

Recording the alignment (e.g. in a separate file):

- owl:Thing
  \[\text{providesAgentRole}\] \rightarrow \text{AgentRole}
  \[\text{performedBy}\] \rightarrow \text{Agent}
  \[\text{hasName}\] \rightarrow \text{xsd:string}

- Movie
  \[\text{hasActor}\] \rightarrow \text{ActorRole}
  \[\text{assumedBy}\] \rightarrow \text{Person}
  \[\text{hasName}\] \rightarrow \text{xsd:string}
The Role Patterns

- range
- domain
- inverse
- range
- domain
- inverse
- existential
- scoped range
- range cardinality
- range cardinality
- existentials
- disjointness
The Agent Role Pattern

Axioms: all previous plus the following.

\[
\begin{align*}
\text{AgentRole} &\subseteq \text{Role} \\
\exists \text{rolePerformedBy}.\text{Agent} &\subseteq \text{AgentRole} \\
\text{AgentRole} &\subseteq \forall \text{rolePerformedBy}.\text{Agent}
\end{align*}
\]
Ontology Axiomatization Support (OWLAx)

- Protégé Plug-In

- Insert class diagram using graphical UI
- System asks you whether to include corresponding axioms (taken from a pool of most common axioms for the diagram)
- You can of course also manually add further axioms.

http://dase.cs.wright.edu/content/ontology-axiomatization-support
Trajectory pattern

- `ssn:Device`
- `time:TemporalEntity`
- `poi:Place`
- `SemanticTrajectory`
- `Fix`
- `Position`
- `Attribute`
- `Segment`

Relations:
- `rdfs:subClassOf` (ssn:Device)
- `hasCreator` (Source)
- `hasLocation` (Position)
- `hasAttribute` (Attribute)
- `hasSegment` (Segment)
- `isTraversedBy` (motionp:MovingObject)
- `startsFrom` (Segment)
- `endsAt` (Segment)
- `atTime` (Fix)
- `nextFix` (Fix)
- `poi:hasSpatialFootprint` (poi:Place)
Trajectory pattern

\[
\text{Fix} \sqsubseteq \exists \text{atTime}. \text{time:TemporalEntity} \land \exists \text{hasLocation}. \text{Position} \\
\land \exists \text{hasFix}^- . \text{SemanticTrajectory}
\]

\[
\text{Segment} \sqsubseteq \exists \text{startsFrom}. \text{Fix} \land \exists \text{endsAt}. \text{Fix} \\
\sqsubseteq 1 \text{startsFrom}. \top \\
\sqsubseteq 1 \text{endsAt}. \top
\]

\[
\text{Segment} \sqsubseteq \exists \text{hasSegment}^- . \text{SemanticTrajectory}
\]

\[
\text{startsFrom}^- \circ \text{endsAt} \sqsubseteq \text{hasNext}
\]

\[
\text{hasNext} \sqsubseteq \text{hasSuccessor}
\]

\[
\text{hasSuccessor} \circ \text{hasSuccessor} \sqsubseteq \text{hasSuccessor}
\]

\[
\text{hasNext}^- \sqsubseteq \text{hasPrevious}
\]

\[
\text{hasSuccessor}^- \sqsubseteq \text{hasPredecessor}
\]
Trajectory pattern

\[
\begin{align*}
  & \text{Fix} \sqcap \neg \exists \text{endsAt}. \text{Segment} \sqsubseteq \text{StartingFix} \\
  & \text{Fix} \sqcap \neg \exists \text{startsFrom}. \text{Segment} \sqsubseteq \text{EndingFix} \\
  & \text{Segment} \sqcap \exists \text{startsFrom}. \text{StartingFix} \sqsubseteq \text{StartingSegment} \\
  & \text{Segment} \sqcap \exists \text{endsAt}. \text{EndingFix} \sqsubseteq \text{EndingSegment} \\
  & \text{SemanticTrajectory} \sqsubseteq \exists \text{hasSegment}. \text{Segment} \\
  & \text{hasSegment} \circ \text{startsFrom} \sqsubseteq \text{hasFix} \\
  & \text{hasSegment} \circ \text{endsAt} \sqsubseteq \text{hasFix} \\
  & \exists \text{hasSegment}. \text{Segment} \sqsubseteq \text{SemanticTrajectory} \\
  & \exists \text{hasSegment}^{-}. \text{SemanticTrajectory} \sqsubseteq \text{Segment} \\
  & \exists \text{hasFix}. \text{Fix} \sqsubseteq \text{SemanticTrajectory} \\
  & \exists \text{hasFix}^{-}. \text{SemanticTrajectory} \sqsubseteq \text{Fix}
\end{align*}
\]
Fig. 1. Partial class diagram of the Trajectory Pattern from [2]. The dashed boxes indicate classes which are themselves (external) patterns, i.e., they need to be specified using a concrete module, or partial ontology.

Fig. 2. Class diagram for the Spatiotemporal Extent pattern.
Axioms

Those inherited from the trajectory pattern, plus

\[
\text{SpatioTemporalExtent} \sqsubseteq \exists \text{hasTrajectory.Trajectory} \\
\text{SpatioTemporalExtent} \sqsubseteq \forall \text{hasTrajectory.Trajectory} \\
\top \sqsubseteq \forall \text{hasSpatioTemporalExtent.SpatioTemporalExtent}
\]
Fig. 3. Example for stationary trajectory: World Chess Championship 2016. The dashed red arrows indicate so-called shortcuts, which are discussed in the text.
Spatiotemp. Extent

Fig. 3. Example for stationary trajectory: World Chess Championship 2016. The dashed red arrows indicate so-called shortcuts, which are discussed in the text.

\[
\begin{align*}
\text{SpatioTemporalExtent}(x) \land \text{hasTrajectory}(x, y) \land \text{hasFix}(y, z) \\
\land \text{StartingFix}(z) \land \text{atTime}(z, w) & \rightarrow \text{hasStartTime}(x, w) \\
\text{SpatioTemporalExtent}(x) \land \text{hasTrajectory}(x, y) \land \text{hasFix}(y, z) \\
\land \text{EndingFix}(z) \land \text{atTime}(z, w) & \rightarrow \text{hasEndTime}(x, w) \\
\text{SpatioTemporalExtent}(x) \land \text{hasTrajectory}(x, y) \land \text{hasFix}(y, z) \\
\land \text{atPlace}(z, w) & \rightarrow \text{hasPlace}(x, w)
\end{align*}
\]
Spatiotemporal Extent
Spatiotemporal event

- SpatioTemporalExtent
  - hasSpatioTemporalExtent
    - Event
      - subEventOf
      - hasInformationObject
        - InformationObject
      - providesParticipantRole
        - ParticipantRole
Spatiotemporal event

\[
\text{subEventOf} \circ \text{subEventOf} \sqsubseteq \text{subEventOf} \\
\exists \text{subEventOf}. T \sqsubseteq \text{Event} \\
T \sqsubseteq \forall \text{subEventOf}. \text{Event} \\
\text{Event} \sqsubseteq \exists \text{hasSpatioTemporalExtent}. \text{SpatioTemporalExtent} \\
T \sqsubseteq \forall \text{hasSpatioTemporalExtent}. \text{SpatioTemporalExtent} \\
\text{Event} \sqsubseteq \exists \text{providesParticipantRole}. \text{ParticipantRole} \\
T \sqsubseteq \forall \text{providesParticipantRole}. \text{ParticipantRole} \\
\text{Event}(x) \land \text{providesParticipantRole}(x, p) \land \text{subEventOf}(x, y) \\
\rightarrow \text{providesParticipantRole}(y, p). \\
\text{Event}(x) \land \text{hasSpatioTemporalExtent}(x, w) \land \text{subEventOf}(x, y) \\
\land \text{Event}(y) \land \text{hasSpatioTemporalExtent}(y, z) \rightarrow \text{subSTEOf}(w, z) \\
T \sqsubseteq \forall \text{hasInformationObject}. \text{InformationObject} \\
\text{AllDisjointClasses} (\text{Event, SpatioTemporalExtent, ParticipantRole, InformationObject})
\]
Quantities and Units

Borrowed from the QUDT ontology
Provenance

Borrowed from PROV-O
Thanks!
References


Adila Krisnadhi, Ontology Pattern-Based Data Integration. Dissertation, Department of Computer Science and Engineering, Wright State University, 2015.

References


Some patterns and their use in the chess ontology

Pascal Hitzler
Data Semantics Laboratory (DaSe Lab)
Data Science and Security Cluster (DSSC)
Wright State University
http://www.pascal-hitzler.de
Worked Example: Chess

- Establish a searchable repository for chess data.

- Starting point are PGN files.
- Should be extendable with other information from
  - Chess websites
  - Wikipedia
  - Geographic data
  - News
  - Etc.

- Use an ontology for information integration.
GeoVoCamps modeling approach

- Collaborative modeling, group ideally has
  - More than one domain experts.
  - People familiar with the base data.
  - People understanding possible target use cases.
  - An ontology engineer familiar with the modeling approach.
  - Somebody who understands formal semantics of OWL.

- Domain experts are queried as to the main notions for the application domain.
  - E.g. for chess, these would include
    - Chess game; move; opening; tournament; players; commentary
GeoVoCamps modeling approach

- From available data and from application use cases, devise competency questions, i.e. questions which should be convertible into queries, which in turn should be answerable using the data.

1. Who played against Kasparov in the round 1994 Lineares tournament? Did (s)he play as a white or black player?
2. What is the first move taken by the black player in the Sicilian Defense opening?
3. Find all games in which Bobby Fischer, playing black, lost in the poisoned pawn variation of the Sicilian Defence opening.
4. Are there any recorded games using the Grünfeld Defence from before the 20th century?
5. What did Kasparov say about his opponent’s first two moves in his commentary about his game against Topalov in the 1999 Tournament in Wijk ann Zee?
6. Who was the first non-Russian world champion after Fischer?
7. Did Bobby Fischer ever play against a grandmaster in Germany?
8. List all world championship games won by forfeit.
• Then prioritize which notions to model first. In the chess case, e.g.

- chess game
- move/half-move
- players
- opening
- tournaments
- commentary
GeoVoCamps modeling approach

- Understand the nature of the things you are modeling.

<table>
<thead>
<tr>
<th>Chess game</th>
<th>...</th>
<th>An Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half-move</td>
<td>...</td>
<td>A Subevent of a chess game</td>
</tr>
<tr>
<td>Player</td>
<td>...</td>
<td>The Role of an Agent</td>
</tr>
<tr>
<td>Opening</td>
<td>...</td>
<td>this is probably complex</td>
</tr>
<tr>
<td>tournaments</td>
<td>...</td>
<td>Events</td>
</tr>
<tr>
<td>commentary</td>
<td>...</td>
<td>this is again more complex</td>
</tr>
</tbody>
</table>
Player as AgentRole

- owl:Thing
  - providesAgentRole
  - AgentRole
  - startsAtTime, endsAtTime
    - TimeInstant
    - performedBy
      - Agent

- BlackPlayerRole
  - rdfs:subClassOf
  - AgentRole
    - providedBy
      - Agent

- WhitePlayerRole
  - rdfs:subClassOf
  - AgentRole
    - performedBy
      - Agent

- ChessGame
  - providesAgentRole
  - AgentRole
    - performedBy
      - Agent
ChessGame as Event

Diagram showing relationships between TemporalExtent, Event, AgentRole, Agent, and Place.
Half-moves
Opening, game result, etc.

We call these “stubs”.

I.e. we’re aware that more fine-grained modeling will be needed for some use cases.

But currently there’s no reason to do it (not in use case, no data), so we only provide “hooks” for future development of the ontology.
Putting things together
Adding commentaries
Adequacy check

• Triplify sample data using the ontology. Does it work?

• Check if competency questions can be answered.

• Add axioms as appropriate (the graph is only for intuition, the OWL axioms are the actual ontology).

• (there are more post-hoc details to be taken care of, but let’s leave it at that)
Axioms

Axioms in this case are mostly straightforward:

- Inherited from Event or AgentRole
- Scoped domain/range restrictions, possibly with some cardinalities
- Basic existentials
- Non-cyclicity of half-move sequence

What about adding, e.g., the following?

ChessGame $\sqsubseteq \geq 0$ subEventOf. ChessTournament

If one of the roles of axiomatization is to improve human understanding of the ontology, then such axioms are helpful!
Shortcuts and Views
Shortcuts

\[
\begin{align*}
\text{ChessGame}(x) \land \text{pAR}(x, y) \land \text{WhitePlayerRole}(y) \land \text{performedBy}(y, z) \\
\land \text{Agent}(z) \land \text{hasName}(z, s) \rightarrow \text{hasWhitePlayer}(x, s)
\end{align*}
\]

\[
\begin{align*}
\text{ChessGame}(x) \land \text{pAR}(x, y) \land \text{BlackPlayerRole}(y) \land \text{performedBy}(y, z) \\
\land \text{Agent}(z) \land \text{hasName}(z, s) \rightarrow \text{hasBlackPlayer}(x, s)
\end{align*}
\]
Translating the rules

However note that the introduction of additional role chains may cause violations of regularity restrictions.
ROWLTab

Modeling OWL with Rules (ROWLTab)

- **Protégé Plug-In**
- **Enter rules using interface very similar to SWRLTab.**
- **But rules are converted into OWL axioms (whenever possible) instead of DL-safe rules.**

E.g., \( \text{Pig}(x) \rightarrow \text{Mammal}(x) \) becomes \( \text{Pig} \sqsubseteq \text{Mammal} \)

and thus carries the correct semantics.

http://dase.cs.wright.edu/content/modeling-owl-rules

We evaluated that ROWL leads to quicker modeling with fewer errors.

http://dase.cs.wright.edu/content/rowl

And see full paper here at ESWC2017
Mapping from Views

We used rules (axioms) to express the mapping from the ontology to the view.

The reverse direction is much more tricky.

\[
\text{ClassA}(x) \land \text{ClassB}(y) \land C_1(x_1) \land \cdots \land C_n(x_n) \land R_1(y_1, y_2) \land \cdots \land R_k(y_k, y_{k+1}) \\
\rightarrow \text{shortcut}(x, y).
\]

\[
\text{shortcut}(x, y) \rightarrow \text{ClassA}(x) \land \text{ClassB}(y) \land \exists x_1 \ldots \exists x_n \exists y_1 \ldots \exists y_n (C_1(x_1) \land \ldots \land C_n(x_n) \land R_1(y_1, y_2) \land \cdots \land R_k(y_k, y_{k+1})).
\]
Mapping from views

Existential rules may be suitable in principle.

\[ \text{shortcut}(x, y) \rightarrow \text{ClassA}(x) \land \text{ClassB}(y) \land \exists x_1 \ldots \exists x_n \exists y_1 \ldots \exists y_n (C_1(x_1) \land \ldots \land C_n(x_n) \land R_1(y_1, y_2) \land \cdots \land R_k(y_k, y_{k+1})) \]

However automated reasoning with the potentially rather complex rule heads requires investigations, in particular if it is to be integrated with ontology reasoning.

A specific case are right-hand-side role chains:

\[ R \sqsubseteq R_1 \circ \cdots \circ R_n, \]
Thanks!
References


Adila Krisnadhi, Ontology Pattern-Based Data Integration. Dissertation, Department of Computer Science and Engineering, Wright State University, 2015.
References


References


OntoPedigree: Design patterns for event-based traceability in provenance-aware supply chains

Monika Solanki
https://w3id.org/people/msolanki
@nimonika
University of Oxford
Motivation

Where are my Things?
Visibility* in supply chains

Visibility is the ability to know exactly where things are at any point in time or where they have been and why.

Enabling Visibility

<table>
<thead>
<tr>
<th>Data/Knowledge Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information and knowledge need to be <strong>interlinked</strong>, <strong>shared</strong> and made available consistently along the supply chain not least for regulatory reasons but also due to increasing consumer demands of being able to <strong>track and trace</strong> commodities.</td>
</tr>
</tbody>
</table>
Pharmaceutical supply chains

Flow of goods and flow of information (Abstraction)
Visibility in Pharmaceutical supply chains

Crucially Important!

Counterfeiting has increasingly become one of the major problems prevalent in these chains. The WHO estimates that between five and eight percent of the worldwide trade in pharmaceuticals is counterfeit.
Pharmaceutical supply chains

GS1 standards* for Visibility

GS1: a neutral, not-for-profit organization dedicated to the design and implementation of global standards and solutions to improve the efficiency and visibility in supply chains.

Core GS1 standards: EPCIS 1.1 & CBV 1.1

GS1 US Secure Supply Chain Task Force: preliminary implementation guidelines* for applying GS1 Standards to U.S. Pharmaceutical supply chains for track and trace.

*http://www.gs1.org/healthcare/standards
*www.gs1us.org/RxGuideline
EPC, EPCIS and CBV

- The Electronic Product Code (EPC)*: provides products with unique, serialised identities.

- Electronic Product Code Information Services (EPCIS v1.1)*: provides a set of specifications for the syntactic capture and informal semantic interpretation of EPC based product information.

- CBV* supplements EPCIS by defining the structure of vocabularies and specific values for the vocabulary elements.

- Events as abstractions for traceability.

*http://www.gs1.org/epcis
SW & LD for Visibility in Supply chains

Research and Application

- Ontological representation of EPCIS events
- OBDA approach towards the transformation of EPCIS RDBMS
- Automated generation of provenance-based traceability artifacts from streaming EPCIS events.
- Identifying and classifying exceptions in events
- Validation of externally acquired traceability artifacts.

monika.solanki@cs.ox.ac.uk, @nimonika
EPCIS(1.1) Events: An informal Intuition

One generic and four specific physical event types

For this talk,

- **EPCISEvent**: the generic EPCIS event.
- **ObjectEvent**: an event that occurred as a result of some action on one or more entities denoted by EPCs.
- **AggregationEvent**: an event that happened to one or more EPC-denoted entities that are physically aggregated.
- **TransactionEvent**: an event in which one or more entities denoted by EPCs become associated or disassociated with one or more identified business transactions.
Data model components

What(product(s)), Where(location), When(time), and Why(business step and disposition) of events (product movement) occurring in any supply chain.

- EPCs
- Time
- Read Points
- Business Location
- Business steps (commissioning, packing, shipping...)
- Disposition (in_transit, retail_sold, returned...)
- Action (ADD, OBSERVE, DELETE)
EEM*: The EPCIS Event Model

Focuses on a tight conformance with the EPCIS 1.1 standard and Simplicity.

Explicitly defines relationships with CBV entities through CBVVocab*.

EEM has been mapped* to PROV-O*.

*http://purl.org/eem#
*www.w3.org/ns/prov-o
*http://purl.org/cbv#
*http://fispace.aston.ac.uk/ontologies/eem_prov.html
EEM Modules

Structure of EEM

Temporal
- time of occurrence
- time of record
- timezone offset

Sensors
- GoodRelations, Collections
- DUL

EPCIS Event
- Object Event
- AggregationEvent
- TransactionEvent
- TransformationEvent
- QuantityEvent*
  * Deprecated in EPCIS 1.1

Business
- Action
- Disposition
- Business step
- Transaction

Spatial
- ReadPoint Location
- Business Location
- Source
- Destination

Product
- EPC
- EPCClass
- Aggregation Identifiers
- Input/Output Quantity List

Semantic Sensor Network Ontology

EPC Readers

CBVVocab
- PROV-O

EPCISException
- Hierarchy of EPCISException

monika.solanki@cs.ox.ac.uk, @nimonika

Event-based traceability in supply chain datasets
The OBDA approach

Pedigrees

- Most widely prevalent in the pharmaceutical industry.
- Pedigree (e-pedigree) is an audit trail that records the path and ownership of a drug as it moves through the supply chain.
- Each stakeholder involved in the manufacture or distribution of the drug adds information to the pedigree.
- “Event-based Linked Pedigrees”: pedigrees based on a relevant subset of the captured EPCIS events.

---

cf. COLD, DeRiVE @ ISWC 2013
Pharmaceutical supply chains

Flow of linked pedigrees (Abstraction)

M. Solanki et al. EPCIS event-based traceability in pharmaceutical supply chains via automated generation of linked pedigrees. ISWC 2014.
Requirements

- Certification and digital signatures
- Product information
- Location information
- Consignment information
- Transaction information
- Partner pedigree information
Competency questions

- Who is the creator of the pedigree?
- What is the supply chain creation status of a given pedigree?
- Which are the business transactions recorded against a particular consignment?
- What are the events associated with pedigrees created between dates X and Y?
- Which products have been shipped together?
- Which other pedigrees are included in the received pedigree?
OntoPedigree: A CO design pattern

Prefixes
ped:http://purl.org/pedigree#
gp:http://purl.org/goodrelations/v1#
eem:http://purl.org/eem#
prov:http://www.w3.org/ns/prov#

monika.solanki@cs.ox.ac.uk, @nimonika

Event-based traceability in supply chain datasets
Linked Pedigree: Axiomatisation

Class: ped:Pedigree
SubClassOf:
  (hasPedigreeStatus exactly 1 ped:PedigreeStatus)
  and (hasSerialNumber exactly 1 rdfs:Literal)
  and (pedigreeCreationTime exactly 1 xsd:DateTime)
  and (prov:wasAttributedTo exactly 1 ped:PedigreeCreator)
  and (ped:hasConsignmentInfo some eem:SetOfEPCISEvents)
  and (ped:hasTransactionInfo exactly 1 eem:SetOfEPCISEvents)
  and (ped:hasProductInfo min 1),
  (prov:wasGeneratedBy only ped:PedigreeCreationService),
  (ped:hasReceivedPedigree only eem:Pedigree),
  prov:Entity
Automated generation of Linked Pedigrees

Streams of EPCIS events, where each EPCIS event is a named graph

Algorithm to extract EPCIS events from streams using INSTANS, an incremental SPARQL query engine

Counterfeit EPC detection as a side-effect of generating linked pedigrees

M. Solanki et al. EPCIS event-based traceability in pharmaceutical supply chains via automated generation of linked pedigrees. ISWC 2014.
Evaluation

EPCIS Event volumes

Data Sources: Sample EPCIS relational data, Grey literature, interviews, surveys, EPCIS experts

Assumption: an average rate of production as 6 days per week and 10 hours per day,

Commissioning events generated based on the number of items ranging from 24,000 to 102,000 per day or approximately 40 to 170 per minute.

Aggregation and shipping events generated considering aggregated items ranging from 100 to 500 (increments of 100) per case and number of cases per pallet ranging from 20 to 100 (increments of 20).

Tumbling window sizes of 3, 5, 7 and 10 hours respectively.
## Evaluation

<table>
<thead>
<tr>
<th>Window size (hrs)</th>
<th>Items/min. event stream velocity</th>
<th>Commissioned events</th>
<th>100-500 per case Aggregation events (increments of 100)</th>
<th>20-100 per pallet Shipping events for each of the aggregates (increments of 20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>120</td>
<td>72000</td>
<td>720/360/240/180/144</td>
<td>36/18/12/9/7 18/9/6/5/4 12/6/4/3/3 18/9/6/5/4 7/4/3/2/2</td>
</tr>
<tr>
<td></td>
<td>170</td>
<td>102000</td>
<td>1020/510/340/255/204</td>
<td>51/26/17/13/11 26/13/9/7/5 17/9/7/5/4 13/7/5/4/3 10/5/4/3/2</td>
</tr>
</tbody>
</table>

Table 1. Number of commissioning, aggregation and shipping events for a window size of 10 hours and item commissioning rate of 120 and 170 per minute
Evaluation: Architecture and Implementation
EPCIS Exceptions

Typical examples

- (e1) Pedigree serial number discrepancy
- (e2) product inference problem - the inability to infer about products contained in an outer container without disaggregation using pedigree information
- (e3) quantity inference problem - the inability to derive the total quantity of items packed in an outer container without disaggregation using pedigree information
- (e4) missing or incorrect containment hierarchy between items and their containers - source of counterfeits.
- (e5) incomplete pedigree data
- (e6) pedigree data with broken chains, i.e., missing intermediate stakeholder pedigree information.
Validation requirements

- Incomplete pedigree: Mandatory information missing
- Pedigree data has broken chain: Pedigrees from other partners are missing
- Pedigree based, receiving and shipping event correlation: EPCs in receiving events do not match the EPCs in the shipping events.
- Temporal validity of shipping and receiving events
- Missing parent-child aggregation
Specifying validation rules

Linked Pedigree \(\xrightarrow{\text{spin:constraint}}\) SPIN Construct Template

- SPIN SPARQL syntax
- **Construct** an exception
- **Where** validation requirements are not satisfied

http://spinnrdf.org/sp.html
http://www.topquadrant.com/spin/tutorial/
Incomplete pedigree

CONSTRUCT
{
_:b0 a eem:PedigreeIncompleteException;
spin:violationRoot ?this;
eem:eventOccurredAt "timeLiteral"xsd:datetime;
eem:associatedBusinessStep cbv:receiving;
....other triples about the exception
rdfs:label "Incomplete pedigree exception".
}

Incomplete pedigree

WHERE
{
  ...
}
FILTER NOT EXISTS { ped:hasPedigreeStatus ?PedigreeStatus;
  ped:hasSerialNumber ?serialNumber;
  ped:pedigreeCreationTime ?pedTime;
  prov:wasAttributedTo ?pedigreeCreator;
  ped:hasConsignmentInfo ?setOfConsEvents;
  ped:hasTransactionInfo ?setOfShipEvents;
  ped:hasProductInfo productInfo.}
Pedigree data has broken chain

CONSTRUCT
{
  _:b0 a eem:BrokenPedigreeChainException;
  ..same as the CONSTRUCT above..
  rdfs:label ‘“Broken pedigree chain exception’
}
WHERE
{
  ?this a ped:Pedigree;
  ped:hasPedigreeStatus ped:IntermediatePedigree;
  ped:hasReceivedPedigree+ ?pedigree.
  FILTER NOT EXISTS {
    ped:hasPedigreeStatus ped:IntermediatePedigree;
    ped:hasReceivedPedigree+ ?pedigree.
  }
}
Receiving and shipping event correlation

SPIN

Do the received EPCs match the shipped EPCs?

Yes

Check the next constraint

No

Throw exception

SPARQL

Receive goods

Pull and dereference pedigrees

Shipping Events and EPCs

Receiving Events and EPCs

start

manufacturer

Pedigree

Warehouse

Pedigree

Shipping

Receiving

28th May 2017, Portoroz

monika.solanki@cs.ox.ac.uk, @nimonika

Event-based traceability in supply chain datasets
Event-based traceability in supply chain datasets
Summary

- Semantic Web standards, ontologies and linked data can be utilised to record and represent real time supply chain knowledge.

- Complex Event Processing over continuous streams of semantically interlinked EPCIS event datasets enable automated generation of linked pedigrees, detection of exceptions and validation of integrity constraints.

- Rule based frameworks can be integrated with distributed realtime computation systems such as Apache Storm to process real time streams of supply chain data.

- The proposed approach is domain independent and can be widely applied to most scenarios of traceability as long as there is conformance to EPCIS 1.1 in the supply chain.
Further information


http://windermere.aston.ac.uk/~monika/ontologies.html
http://windermere.aston.ac.uk/~monika/publication.html
Modular Ontology Design and Use Case: The GeoLink Example

Pascal Hitzler
Data Semantics Laboratory (DaSe Lab)
Data Science and Security Cluster (DSSC)
Wright State University
http://www.pascal-hitzler.de
This Tutorial (all parts)

- Pascal Hitzler (60 mins): *Introduction and first examples*
- Monika Solanki (30 mins): *Example “modeling vaccine traceability”*

**coffee**

- Pascal Hitzler (60 mins): *Example “GeoLink Modular Ontology”*
- Agnieszka Lawrynowicz (30 mins): *Example “Reporting Event ODP”*

**lunch**

- Karl Hammar with all others (3h): *Hands-on, the WebProtege XDP plug-in*
The NSF EarthCube Program: Developing a Community-Driven Data and Knowledge Environment for the Geosciences

“concepts and approaches to create integrated data management infrastructures across the Geosciences.”

“EarthCube aims to create a well-connected and facile environment to share data and knowledge in an open, transparent, and inclusive manner, thus accelerating our ability to understand and predict the Earth system.”
EarthCube GeoLink Scenario

GeoLink: An EarthCube “Building Block” project (2014-2017)

How to realize data search across many large-scale geoscience data repositories, such that

- The approach is extendable to new repositories.
- The scope can extend across all of the Geosciences.
- The search capabilities can be made more fine-grained in the future if desired.

Central idea: Use a modular, extendable ontology for the integration of metadata.
GeoLink

An interactive demonstration of the integrated GeoLink data is available at

http://demo.geolink.org

At http://www.geolink.org/ there are links to the complete schema, a SPARQL Endpoint, publications, etc.
The GeoLink Framework

GeoLink Ontology Patterns

- **isAssociatedWithCruise**: relates a cruise to a specific event or place.
- **isResultOf**: indicates the result of a sampling process.
- **hasSpatialFootprint**: connects a place with its spatial extent.
- **Geometry**: represents the spatial properties of a place.
- **Publish**: relates a publication to a dataset.
- **Dataset**: contains information about a digital object.
- **Digital Object**: represents a piece of information.
- **Event**: describes a temporal occurrence.
- **Sampling Process**: involves the collection of samples.
- **Physical Sample**: captures the essence of a sample.
- **Property**: describes a characteristic of an object.
- **Organization**: represents an institution.
- **Agent**: includes person, vessel, and other entities.
- **Funding Award**: supports the activities.
- **Alignment(s)**: integrate different datasets.

Datasets: BCO-DMO, DataOne, IEDA, IODP, LTER, MBLWHOI Library, R2R.
High-level overview of the GeoLink Modular Ontology (GMO). Each box stands for a module, which has been modeled in its own right.
Identifier pattern

\[ \text{Identifier} \sqsubseteq (\leq 1 \text{hasIdentifierScheme}.(\text{xsd:anyURI} \sqcup \text{xsd:string})) \]
\[ \text{Identifier} \sqsubseteq (\geq 1 \text{hasIdentifierValue}.\text{xsd:string}) \]
\[ \exists \text{hasIdentifierScheme}.(\text{xsd:anyURI} \sqcup \text{xsd:string}) \sqsubseteq \text{Identifier} \]
\[ \exists \text{hasIdentifierValue}.\text{xsd:string} \sqsubseteq \text{Identifier} \]
\[ \text{Identifier} \sqsubseteq \forall \text{hasIdentifierScheme}.(\text{xsd:anyURI} \sqcup \text{xsd:string}) \]
\[ \text{Identifier} \sqsubseteq \forall \text{hasIdentifierValue}.\text{xsd:string} \]
Ontology Axiomatization Support (OWLAX)

• Protégé Plug-In

• Insert class diagram using graphical UI
• System asks you whether to include corresponding axioms (taken from a pool of most common axioms for the diagram)
• You can of course also manually add further axioms.

http://dase.cs.wright.edu/content/ontology-axiomatization-support
Axioms – Systematically

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>$A \cap B \subseteq \bot$</td>
<td>6.</td>
</tr>
<tr>
<td>2.</td>
<td>$\exists R. T \subseteq A$</td>
<td>7.</td>
</tr>
<tr>
<td>3.</td>
<td>$\exists R. B \subseteq A$</td>
<td>8.</td>
</tr>
<tr>
<td>4.</td>
<td>$T \subseteq \forall R . B$</td>
<td>9.</td>
</tr>
<tr>
<td>5.</td>
<td>$A \subseteq \forall R . B$</td>
<td>10.</td>
</tr>
</tbody>
</table>

1. $A$ DisjointWith $B$
2. $R$ some owl:Thing SubClassOf $A$
3. $R$ some $B$ SubClassOf $A$
4. owl:Thing SubClassOf $R$ only $B$
5. $A$ SubClassOf $R$ only $B$
6. $A$ SubClassOf $R$ some $B$
7. $B$ SubClassOf inverse $R$ some $A$
8. owl:Thing SubClassOf $R$ max 1 owl:Thing
9. owl:Thing SubClassOf $R$ max 1 $B$
10. $A$ SubClassOf $R$ max 1 owl:Thing
11. $A$ SubClassOf $R$ max 1 $B$
12. owl:Thing SubClassOf inverse $R$ max 1 owl:Thing
13. owl:Thing SubClassOf inverse $R$ max 1 $A$
14. $B$ SubClassOf inverse $R$ max 1 owl:Thing
15. $B$ SubClassOf inverse $R$ max 1 $A$

(disjointness)
(domain)
(scoped domain)
(range)
(scoped range)
(existential)
(inverse existential)
(functionality)
(qualified functionality)
(scoped functionality)
(qualified scoped functionality)
(inverse functionality)
(inverse qualified functionality)
(inverse scoped functionality)
(inverse qualified scoped functionality)
OWL Ax Protégé plug-in

In: Proc. ISWC 2016 poster & demos
http://dase.cs.wright.edu/content/ontology-axiomatization-support
Information Object pattern

\[ T \subseteq (\leq 1 \text{ describedBy } . \text{InformationObject}) \]
\[ \text{InformationObject} \subseteq (\leq 1 \text{ describedBy}^-. T) \]
\[ \text{InformationObject} \subseteq \neg \exists \text{describedBy}. \text{InformationObject} \]
\[ \exists \text{hasWebpage}. \text{xsd:anyURI} \subseteq \text{InformationObject} \]
\[ \exists \text{alsoKnownAs}. \text{xsd:string} \subseteq \text{InformationObject} \]
\[ \exists \text{hasCanonicalName}. \text{xsd:string} \subseteq \text{InformationObject} \]
\[ \exists \text{hasDescription}. \text{xsd:string} \subseteq \text{InformationObject} \]
Alignment to external ontologies or vocabularies, rather than direct reuse:

```
geo:Geometry ⇔ owl:equivalentClass ⇔ Geometry
```

GeoSPARQL, http://www.opengis.net/ont/geosparql
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
Good axioms

Specificity matters: Problems with domain/range.

Recommendations often heard (but are problematic):

- Indicate domain and range for your properties.
- Reuse as many existing vocabularies as you can.

But there are problems with this:

**Ontology 1:**

![Diagram showing foaf:name as domain of Human and xsd:string as range]

- Human → domain(foaf:name) = Human

**Ontology 2:**

![Diagram showing foaf:name as domain of Organization and xsd:string as range]

- Organization → domain(foaf:name) = Organization

Logical consequence after merge:  
Human \( \equiv \) Organization
Recommendations

- Make rich axiomatizations
- Avoid re-use of external vocabularies (rather provide an additional file with mappings for those who want to use it)
- Avoid naïve domain and range axioms.

Alternative to naïve domain/range: scoped domain and range.

\[ A(x) \land p(x, y) \rightarrow B(y) \]  scoped range
\[ B(y) \land p(y, x) \rightarrow A(x) \]  scoped domain

both rules can be expressed in OWL.
Place shortcuts

```
Place(x) \land io:describedBy(x, y) \land io:hasCanonicalName(y, z) \rightarrow hasPlaceName(x, z)
Place(x) \land io:describedBy(x, y) \land io:alsoKnownAs(y, z) \rightarrow hasPlaceAliasName(x, z)
Place(x) \land io:describedBy(x, y) \land io:hasDescription(y, z) \rightarrow hasTextDescription(x, z)
Place(x) \land io:describedBy(x, y) \land io:hasWebpage(y, z) \rightarrow hasWebpage(x, z)
```
Funding Award pattern

- **AgencyProgramManagerRole**
- **SponsorRole**
- **ar:AgentRole**
- **CoPrincipalInvestigatorRole**
- **PrincipalInvestigatorRole**
- **AwardAmount**
  - hasCurrency
  - hasCurrencyValue
  - hasAwardAmount
  - startsOnDate, endsOnDate
- **FundingAward**
  - startsOnDate, endsOnDate
  - io:InformationObject
- **Currency**
- **xsd:decimal**
- **io:InformationObject**
Program shortcuts

Note: shortcut does not go to data value.
Organization pattern

- ar:AgentRole
  - ar:providesAgentRole
  - rdfs:subClassOf
  - AffiliationRole

- Organization
  - rdfs:subClassOf
  - io:describedBy
  - hasSite
  - Place

- io:InformationObject
  - haSubOrganization

May 2017 – ESWC 2017, Portoroz, Slovenia – Pascal Hitzler
**Person**

```
ar:Agent
   rdfs:subClassOf
Person
   hasPersonallNfoltem
PersonAllNfoltem
   hasLiteralValue
   rdfs:Literal

Instant
   startsAtTime,
   endsAtTime

owl:Thing
   hasValue
```

---

May 2017 – ESWC 2017, Portoroz, Slovenia – Pascal Hitzler
Person Name

\[ \text{Person} \sqsubseteq \text{ar:Agent} \]
\[ \text{PersonalInfoltem} \sqsubseteq (=1 \text{ hasPersonalInfoltem} . \text{Person}) \]
\[ \text{PersonalInfoltem} \sqsubseteq (=1 \text{ startsAtTime. Instant}) \]
\[ \exists \text{hasPersonalInfoltem}. \text{PersonalInfoltem} \sqsubseteq \text{Person} \]
\[ \exists \text{startsAtTime. Instant} \sqsubseteq \text{PersonalInfoltem} \]
Platform pattern (stub)

Platform \(\xrightarrow{\text{io:describedBy}}\) io:InformationObject
Property Value pattern

```
owl:Thing

hasProperty

PropertyKind

hasPropertyKind

Property

hasPropertyUnit

PropertyKind

hasPropertyKind

hasPropertyName

xsd:string

hasPropertyValue

rdfs:Literal

Property

hasPropertyValue

PropertyUnit

hasPropertyUnit
```
Vessel module

\[ \text{pf:Platform} \rightarrow \text{ar:AgentRole} \rightarrow \text{VesselOwnerRole} \]

\[ \text{Vessel} \rightarrow \text{io:InformationObject} \rightarrow \text{VesselDraft} \rightarrow \text{VesselDisplacement} \rightarrow \text{VesselBeam} \rightarrow \text{VesselLength} \]

\[ \text{pv:Property} \rightarrow \text{pv:hasPropertyKind} \rightarrow \text{VesselDraft} \rightarrow \text{VesselDisplacement} \rightarrow \text{VesselBeam} \rightarrow \text{VesselLength} \]

\[ \text{rdfs:subClassOf} \rightarrow \text{ar:providesAgentRole} \rightarrow \text{io:describedBy} \]
Vessel module

Vessel \sqsubseteq pf:Platform

Vessel \sqsubseteq (\exists 1 \: io:describedBy.io:InformationObject)

Vessel \sqsubseteq \exists ar:providesAgentRole. VesselOwnerRole

Vessel \sqsubseteq \exists pv:hasProperty. VesselBeam

Vessel \sqsubseteq \exists pv:hasProperty. VesselDisplacement

Vessel \sqsubseteq \exists pv:hasProperty. VesselDraft

Vessel \sqsubseteq \exists pv:hasProperty. VesselLength

VesselBeam \sqsubseteq \exists pv:hasPropertyKind. \{VesselBeam\}

VesselDisplacement \sqsubseteq \exists pv:hasPropertyKind. \{VesselDisplacement\}

VesselDraft \sqsubseteq \exists pv:hasPropertyKind. \{VesselDraft\}
Vessel shortcuts

- pf:Platform
- ar:Agent
- ar:AgentRole
- rdfs:subClassOf
- VesselOwnerRole
- ar:providesAgentRole
- Vessel
- io:describedBy
- io:InformationObject
- io:hasPrimaryIdentifier
- id:Identifier
- hasVesselOwner
- hasICESPlatformCode
- hasVesselBeam, hasVesselDisplacement, hasVesselDraft, hasVesselLength
- pv:hasProperty
- pv:hasPropertyValue
- pv:Property
- rdfs:subClassOf
- hasPropertyValue
- rdfs:Literal
- rdfs:subClassOf
- VesselDraft
- VesselDisplacement
- VesselBeam
- VesselLength
- xsd:string
- id:hasIdentifierScheme
- "ICES"
- id:hasIdentifierValue
- rdfs:subClassOf
- rdfs:Literal
Cruise module

Cruise $\subseteq$ e:Event

Cruise $\subseteq$ (\(=1\) hasTrajectory. CruiseTrajectory)

Cruise $\subseteq$ (\(=1\) undertakenBy. v:Vessel)

Cruise $\subseteq$ (\(=1\) io:describedBy.io:InformationObject)

CruiseTrajectory $\subseteq$ t:Trajectory

ScientistRole
SchedulerRole
ChiefScientistRole
CoChiefScientistRole
EducatorRole
ObserverRole
OperatorRole
StudentRole
TechnicianRole
Cruise module

\[
\text{CruiseTrajectory} \subseteq \exists t: \text{hasFix}.(t: \text{StartingFix} \land \exists t: \text{hasNextFix}.t: \text{Fix})
\]

\[
v: \text{Vessel} \subseteq t: \text{MovingObject}
\]

\[
\text{Port} \subseteq \text{pl: Place}
\]

\[
\text{hasTrajectory} \circ \text{undertakenBy} \subseteq \text{traveledBy}
\]

\[
t: \text{hasPosition}(x, y) \land p: \text{hasSpatialFootprint}(z, y) \land \text{Port}(z) \rightarrow \text{atPort}(x, z)
\]
Cruise shortcuts

id:Identifier

xsd:string

id:hasIdentifierValue

io:InformationObject

io:describedBy

ar:AgentRole

hasScientist, hasChiefScientist, hasCoChiefScientist, hasScheduler, hasEducator, hasObserver, hasOperator, hasStudent, hasTechnician

ar:performedBy

ar:Agent

hasRelatedCruise

rdfs:subClassOf

hasCruiseID

undertakenBy

hasTrajectory

e:occursAtTime

Cruise

e:Event

pl:Place

hasPortCall, hasStartPortCall, hasEndPortCall, e:occursAtPlace

rdfs:subClassOf

Port

pl:hasSpatialFootprint

pl:hasSpatialFootprint

atPort

t:hasPosition

t:hasFix

v:Vessel

rdfs:subClassOf

t:MovingObject

traversedBy

t:Segment

traveledBy

t:EndingFix

hasPortCall

hasEndPortCall

hasStartPortCall

e:occursAtPlace

e:occursAtPlace

t:hasPosition

t:Temporalentity

t:atTime

CruiseTrajectory

hasCruiseID

traveledBy

t:Segment

traversedBy

t:MovingObject

t:StartingFix

t:Fix

t:hasFix

atPort

t:hasPosition
What we need

• A critical amount of simple, general-purpose patterns
  – Well-documented
  – Not too generic, not too specialized
  – Interrelated (e.g., different versions with different granularity of the same notion)

• Languages for describing patterns.

• Languages for describing modular ontologies based on patterns.

• Tools for working directly with patterns in ontology engineering (see afternoon session – Karl Hammar’s work)
Thanks!
References


Adila Krisnadhi, Ontology Pattern-Based Data Integration. Dissertation, Department of Computer Science and Engineering, Wright State University, 2015.
References


The Reporting Event ODP

Agnieszka Ławrynowicz¹, Ewa Kowalczuk¹

¹Faculty of Computing, Poznan University of Technology, Poznan, Poland

May 28, 2017
Tutorial: Modular Ontology Modeling with Ontology Design Patterns at ESWC2017
emerging need for storing and investigating not only information about a particular event, but also the provenance of the information and circumstances of its provision

properties attributed to an event are not stored as facts, but as a narrative of a particular agent, which could differ from the narratives of other agents
Do existing event ontologies address the subjectivity of event properties? 1/2

- **CIDOC Conceptual Reference Model (CRM):** does not allow to mark the level of property value probability or attribute a property assignment to a particular agent.

- **Linking Open Descriptions of Events ontology (LODE):** allows for linking events to media objects presenting them and thus denoting sources.

- **IPTC NewsML-G2 controlled vocabulary:** allows to define whether an event and its time interval are confirmed or not; **IPTC rNews model** introduces a property `rnews:accountablePerson` for linking to a person responsible for a particular news item (as a whole).
Do existing event ontologies address the subjectivity of event properties? 2/2

- **Simple Event Model (SEM):** introduces a subclass of `sem:Constraint`, called `sem:View`, allowing to mark some attributed property as a belief (point of view) of a particular `sem:Authority`. The property assignment is constant: there is no means of representing the fact that a view changed over time.

- **BBC Storyline Ontology** introduces a notion of `nsl:Storyline`, to denote the editorial perspective of an event or a group of events. It can be attributed to a specific owner. Storylines have a larger span than a single event. They can include `nsl:StorylineSlots`: real world events or inner storylines.
To allow for modelling situations in which the knowledge about an event cannot be treated as certain. It is particularly useful for cases in which two or more agents provide different, contradictory information about the same event. Also for modelling situation in which a single agent provided contradictory information about the same event at different points in time. The pattern allows for stating different circumstances of an act of the information provision.
Competency questions

- What characteristics (e.g. date, participants, cause) is an actual event said to have?
- Which agent made a statement about an actual event?
- On which sources these statements were based?
- What were the circumstances of providing information about an actual event?
Pattern formalization

ActualEvent ∈ Event
ActualEvent ∈ SocialObject ∈ Object
ActualEventView ∈ Event
ActualEventView ∈ InformationObject ∈ Object
ActualEventView ∈ = 1 isAbout.ActualEvent
ActualEventView ∈ ∀ isAbout. ActualEvent
ActualEventView ∈ = 1 reports⁻¹. ReportingEvent
ActualEventView ∈ ∀ hasContext. ActualEventViewContext
ActualEventViewContext ∈ ∃ hasContext⁻¹. ActualEventView
ReportingEvent ∈ Event
ReportingEvent ∈ Situation
ReportingEvent ∈ ∃ reports. ActualEventView
ReportingEvent ∈ ∀ reports. ActualEventView
ReportingEvent ∈ ∃ hasAuthor. EventReporter
ReportingEvent ∈ ∀ isBasedOn. Source
EventReporter ∈ Agent
EventReporter ∈ ∃ hasAuthor⁻¹. ReportingEvent
Source ∈ (Event △ Object)
reports ∈ isSettingFor
Intent: To represent the meaning of an information object: the concepts it expresses, the things it is about.

Competency Questions:
- What is the meaning of an information object?
- What information objects express this meaning?
- What is this about?
- How can I call this?
http://ontologydesignpatterns.org/wiki/Submissions:IntensionExtension
**Intent:** To represent contexts or situations, and the things that are contextualized.

**Competency Questions:**
- What is the context or situation of something?
- What are the things present in this context or situation?
http://ontologydesignpatterns.org/wiki/Submissions:S Situation
Sample use of Reporting Event ODP: Historical Debate
Can be used for modelling situations in which we are not certain that a particular actual event has the properties which were described in a news message. We want to define the properties of an actual event which were reported (time, place, actors, subevents, cause, effect etc.), but not to treat them as universal, verified knowledge. The pattern also allows to define who is responsible for a particular description of an event and how this description is dealt with.
What aspects of an actual event were presented in the news message?

Who reported an actual event? Which news provider they represented?

When was a certain actual event reported for the first time?

What actual events are presented in a certain medium/by media of a certain news provider?

How was an actual event presented?
News Reporting Event ODP

http://ontologydesignpatterns.org/wiki/Submissions:NewsReportingEvent
Pattern formalization

NewsReportingEvent ⊑ ReportingEvent
NewsReportingEvent ⊑ TimeIndexedSituation
NewsReportingEvent ⊑ ∃hasAuthor. NewsEventReporter
NewsReportingEvent ⊑ ∃presentedAt. Media
NewsEventReporter ⊑ EventReporter
NewsEventReporter ⊑ ∃hasAuthor⁻¹. NewsReportingEvent
NewsProvider ⊑ SocialAgent
NewsProvider ⊑ ∃owns. Media
Media ⊑ = 1 owns⁻¹. NewsProvider
NewsPresentationContext ⊑ ∃hasPresentationContext⁻¹. NewsReportingEvent
hasPresentationContext ⊑ isSettingFor
Acting For ODP

**Intent:** To represent that some agent is acting in order to forward the action of a social (non-physical) agent.

**Competency Questions:**
- Who is working for which organization?
- Who is representing the company?

http://ontologydesignpatterns.org/wiki/Submissions:ActingFor
**Intent:** To represent time indexed situations.

**Competency Questions:**
- At what time did a certain situation occur?
- What situations occurred at a certain time?
http://ontologydesignpatterns.org/wiki/Submissions:TimeIndexedSituation
Sample use of News Reporting Event ODP: Presentation of Social Unrests
Thank you!
eXtreme Design

- "a family of methods and associated tools, based on the application, exploitation, and definition of Ontology Design Patterns (ODPs) for solving ontology development issues" – Presutti et al.
- Agile, iterative, pair development, testing emphasis
- Requirements written as user stories formalised as Competency Questions, Contextual Statements, Reasoning Requirements
- Tight customer integration
- Key steps: find ODP, instantiate ODP, integrate solution
Project initiation and scoping

Identify CODP catalogues

Collect requirement stories

Select story

Elicit requirements

Select set

Match and select ODPs

Instantiate and integrate ODPs

Test module

All req:s covered?

No

Yes

Release module

Integrate partial solutions, evaluate, revise

Release new version

All stories covered?

No

Yes
XD for WebProtégé (XDP)

Fork of WebProtégé including tooling to support some XD steps:

- Find ODPs
- Instantiate ODPs (template-based or specialisation-based)
- Integrate ODPs into solution (basic alignment)

Also includes visualization, courtesy of code from the VisualDataWeb project and new UI tabs for advanced editing

Some restrictions of WebProtégé:

- No reasoning
- ODP namespaces cloned, not imported
**Graphical representation**

```
owl:Thing
```

```
itemContent : not (Item)[1..1] 
itemOf : Bag

Bag

BasicPlan 
BasicPlanExecution 
CatchRecord 
Classification 
ClimaticZone 
Co-participation 
collection entity
CommunicationEvent 
Communities
```

**General description**

**Name**: Bag

**Intent**: To model bags of items (elements). The Bag is characterized by a collection that can have multiple copies of each object.

**Solution description**: The Bag is characterized by a collection that can have multiple copies of each object. This is performed through the Item entity. The Item is linking exactly one resource through the relationship itemContent.
Select the appropriate Content Ontology Design Pattern instantiation method from the choices below. For a discussion on their respective attributes and effects, see http://goo.gl/dv8pA3

**Template-Based Instantiation**

In this method the CODP building block is treated as a template that is instantiated into the target ontology module by way of copying and renaming its constituent classes and properties. Advantages of this method include that CODP-level generic concepts that may be off-putting to less experienced modellers are not included in the final ontology, but only the CODP structure is kept. Disadvantages include that future alignment to other ontologies using the same CODPs may be complicated, as the IRIs of COPD-level concepts are not kept.

**Import-Based Instantiation**

In this method the original CODP is imported into the target ontology module, and instantiation is performed via specialization of CODP classes and properties using subsumption axioms. Advantages of this method include increased traceability and ease of alignment with other CODPs, as IRIs of CODP-level concepts are maintained.
Please provide labels for the ODP entities below that make sense when adapting the ODP to your domain.

**Classes**

| item                      | ==> | My item class |
| (collections) Bag         | ==> | My bag class  |

**Object Properties**

| item content              | ==> | my item has some content |
| item of                   | ==> | is item in my bag        |
| has item                  | ==> | my bag has my item       |
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Prefix: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>
Prefix: rdfs: <http://www.w3.org/2000/01/rdf-schema#>

Ontology: <wtmp:entity>

ObjectProperty: <wtmp:entity#is item in my bag>
  Domain:
   <wtmp:entity#My item class>
  Range:
   <wtmp:entity#My bag class>

ObjectProperty: <wtmp:entity#my item has some content>
  Domain:
   <wtmp:entity#My item class>

ObjectProperty: <wtmp:entity#my bag has my item>
  Domain:
   <wtmp:entity#My bag class>
Hands-On Session

- Using XDP and ODPs, construct an ontology covering a set of requirements and structuring a set of provided data, in the policing domain.
- Inspired by a real-world project and real-world data.
- Goal: try out and learn about the method, the tooling, and look at some ODPs.
CAVEATS

• ODP Portal, tooling, etc. are mirrored from the Internet, to account for the lack of Internet connectivity. Some IRIs you see here do not exist in the real world, or lead to content that is not the same as what you see here.

• Further: ODP quality varies greatly in the real world: stale IRIs, bad documentation, bad illustrations, dependency on remote references, etc.

• Thus: four specifically suggested patterns have been tampered with in order to simplify modelling. E.g., merged import closure, added missing documentation, added some common-sense assumptions.

• Finally: this is beta-quality research software. Expect some bugs.
Get started

Poll: Who wants some Google Refine/OpenRefine introduction as well?

• WiFi SSID “ODP Tutorial”, password: “eswc2017”
• Instructions: http://ontologydesignpatterns.org/instructions.txt
• Data: http://ontologydesignpatterns.org/data.zip
• XDP Instance: http://wp.xd-protege.com
• ODP Portal: http://ontologydesignpatterns.org
• WebVOWL Instance: http://vis.xd-protege.com