



Short paper

A Web-based UI to Enable Semantic Modeling for Everyone

André Pomp, Alexander Paulus, Daniel Klischies, Christian Schwier, Tobias Meisen

SEMANTiCS 2018 Vienna, Austria

Alexander Paulus, M.Sc.

Institute of Information Management in Mechanical Engineering

RWTH Aachen University

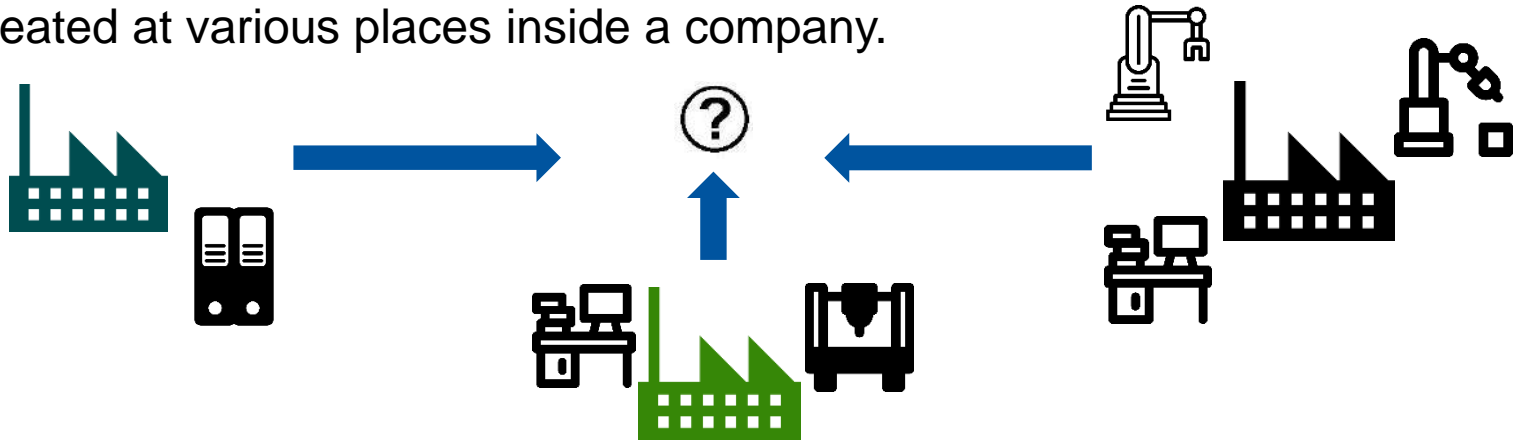
Germany

Motivation

Handling Heterogeneous Data

Distributed Data Sources

Data is created at various places inside a company.



- Heterogenous Provision:



- Heterogenous Formats:



- Heterogenous Models:



- Heterogenous Representations and Meanings:  °C °F K

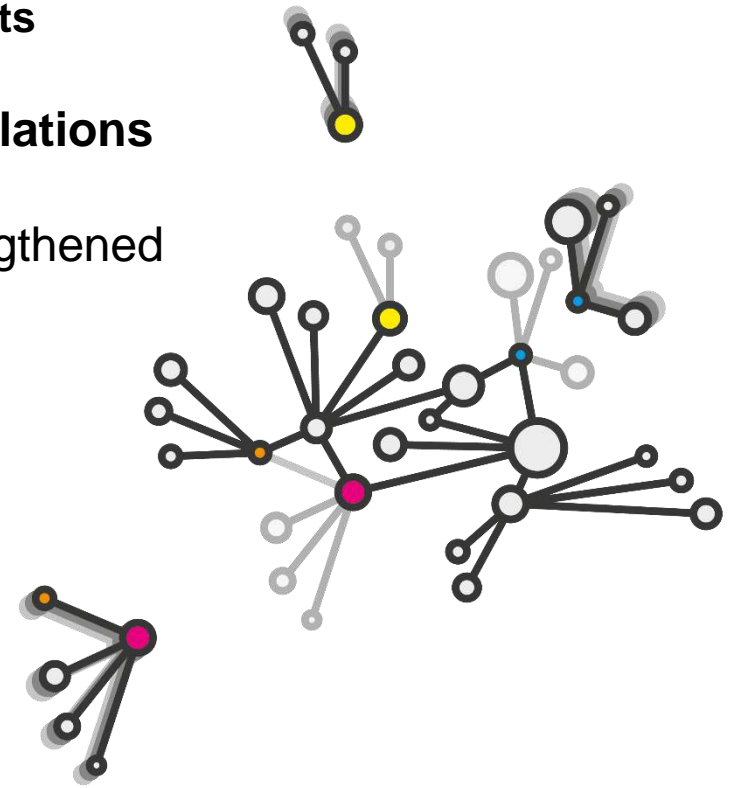
ESKAPE is a company-wide data marketplace, which...

- ... can deal with **heterogeneous data** (in terms of format, representation, etc.)
- ... uses **semantic models** based on a **dynamic semantic knowledge graph** to **describe the information of the data**
- ... has a **knowledge graph** which **evolves incrementally** based on **the provided data** and their **semantic models**
- ... enables **search, retrieval** and **enrichment** of **integrated data**



Extension of knowledge graph

- **New semantic models** are **added / merged** into the knowledge graph
 - **Reused concepts** serve as anchor points
 - Probabilities help **assigning relations to concepts**
- External sources help **validate concepts** and **relations**
- Knowledge graph is **constantly growing** / strengthened
- Semantic models allow users to model their sources **freely** and add concepts during this process



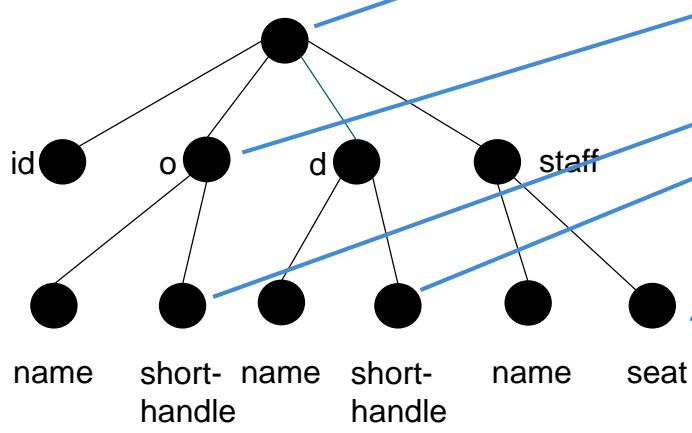
Small example data schema

```
{
  "id": "AL0025-20200123-1456-1634",
  "o": {"name": "Cologne", "shorthand": "CGN"},
  "d": {"name": "Berlin", "shorthand": "BER"},
  "staff": [
    {"name": "Pilot 1", "seat": "P1"},
    {"name": "Pilot 2", "seat": "P2"},
    {"name": "Flight attendant 1", "seat": "FA1"},
    {"name": "Flight attendant 2", "seat": "FA2"}
  ]
}
```

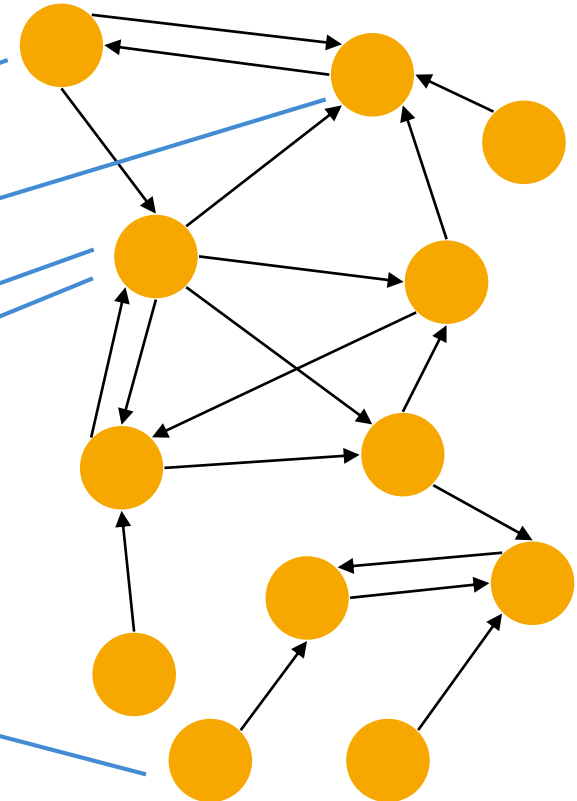
Motivation

Creating Semantic Models

```
{  
  "id": "AL0025-20200123-1456-1634",  
  "o": {"name": "Cologne", "shorthand": "CGN"},  
  "d": {"name": "Berlin", "shorthand": "BER"},  
  "staff": [  
    {"name": "Pilot 1", "seat": "P1"},  
    {"name": "Pilot 2", "seat": "P2"},  
    {"name": "Flight attendant 1", "seat": "FA1"},  
    {"name": "Flight attendant 2", "seat": "FA2"}  
  ]  
}
```



Knowledge Graph



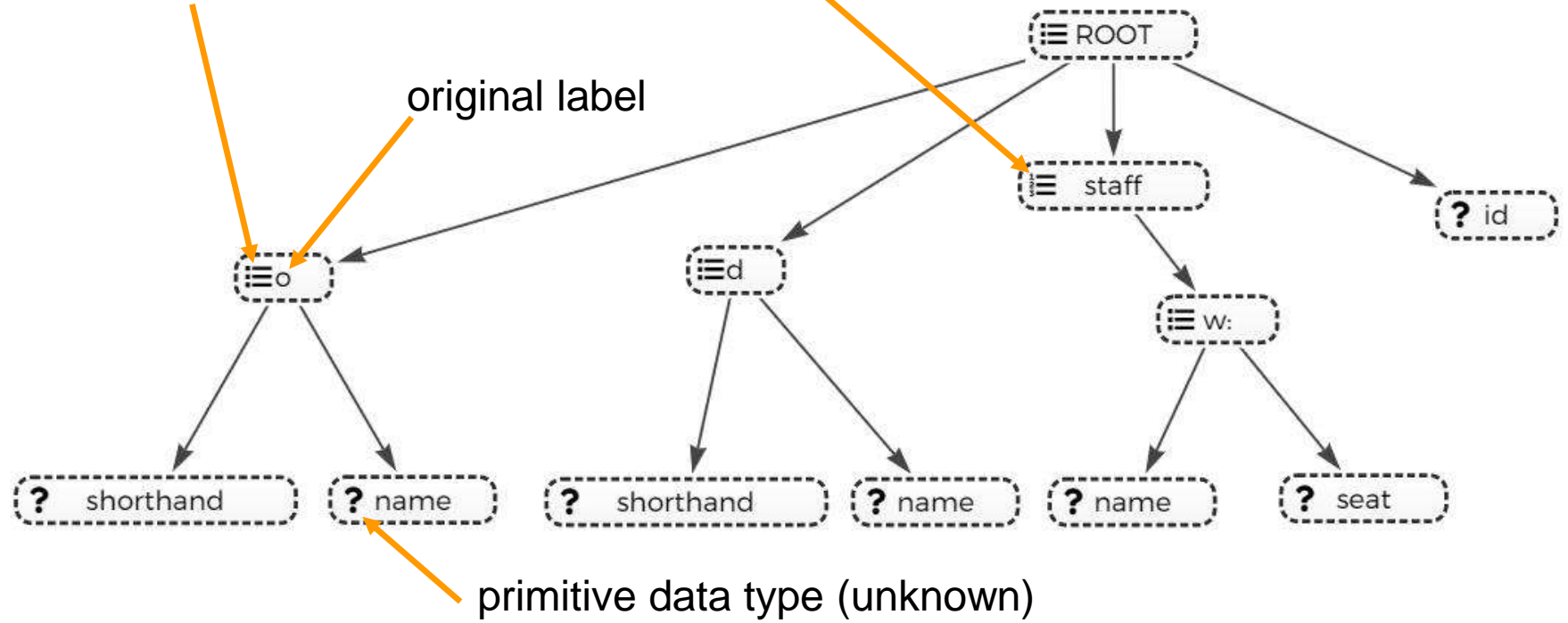
Display the detected data schema

Start from user's point of view

data type indicator (list)

data type indicator (object)

original label



primitive data type (unknown)

Display the detected data schema

Clean and slim GUI for unexperienced users

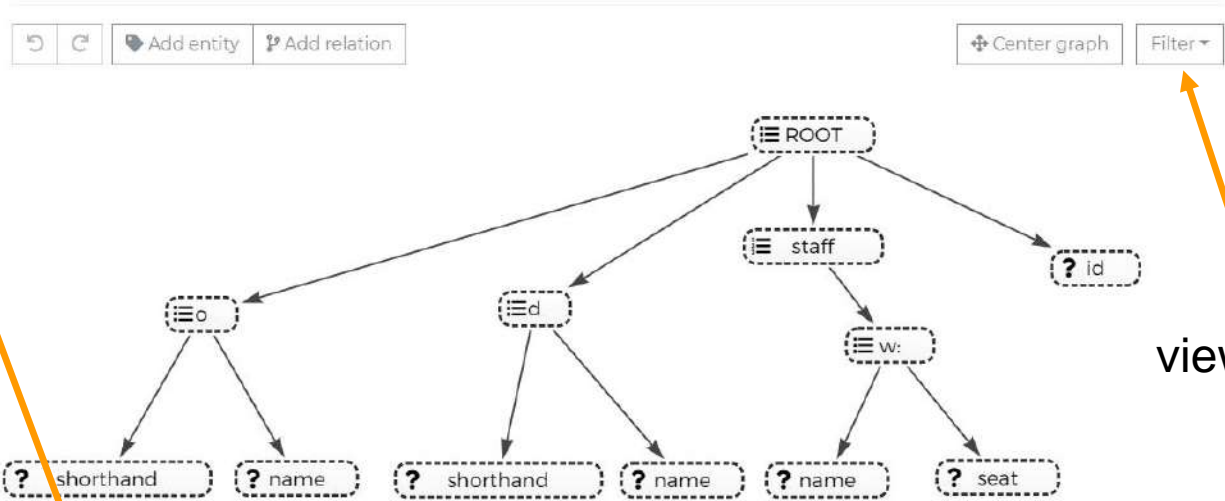
list of available entities

Entities Relations

+ New Filter...

Enter a search term to find matching concepts.

- abstract
- acm identifier
- activity
- actor
- address

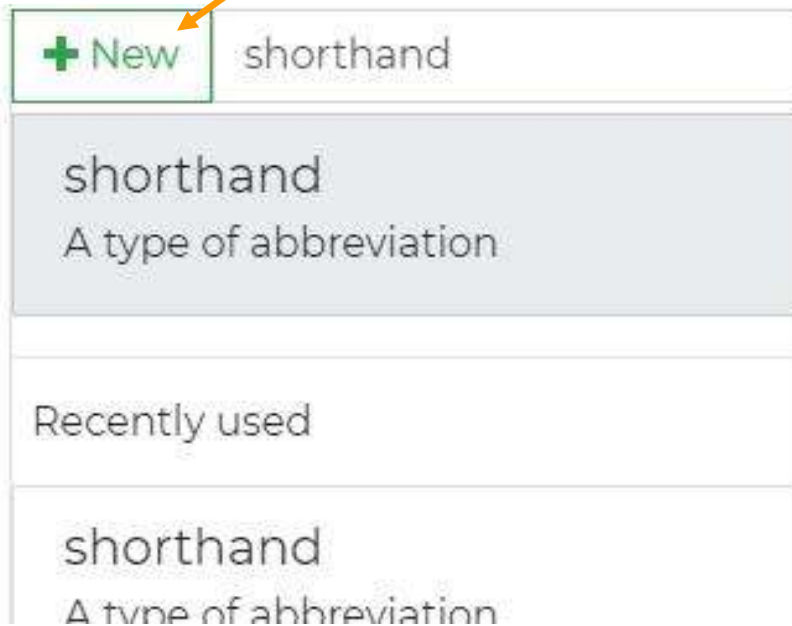


switch to relations

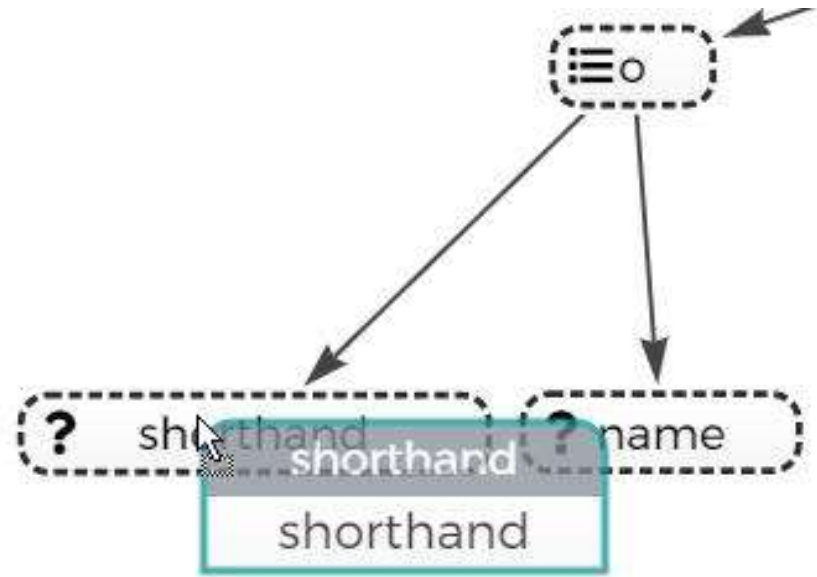
view toggles

Use local and global semantic networks to offer concepts

create new entities / concepts and relations on the fly

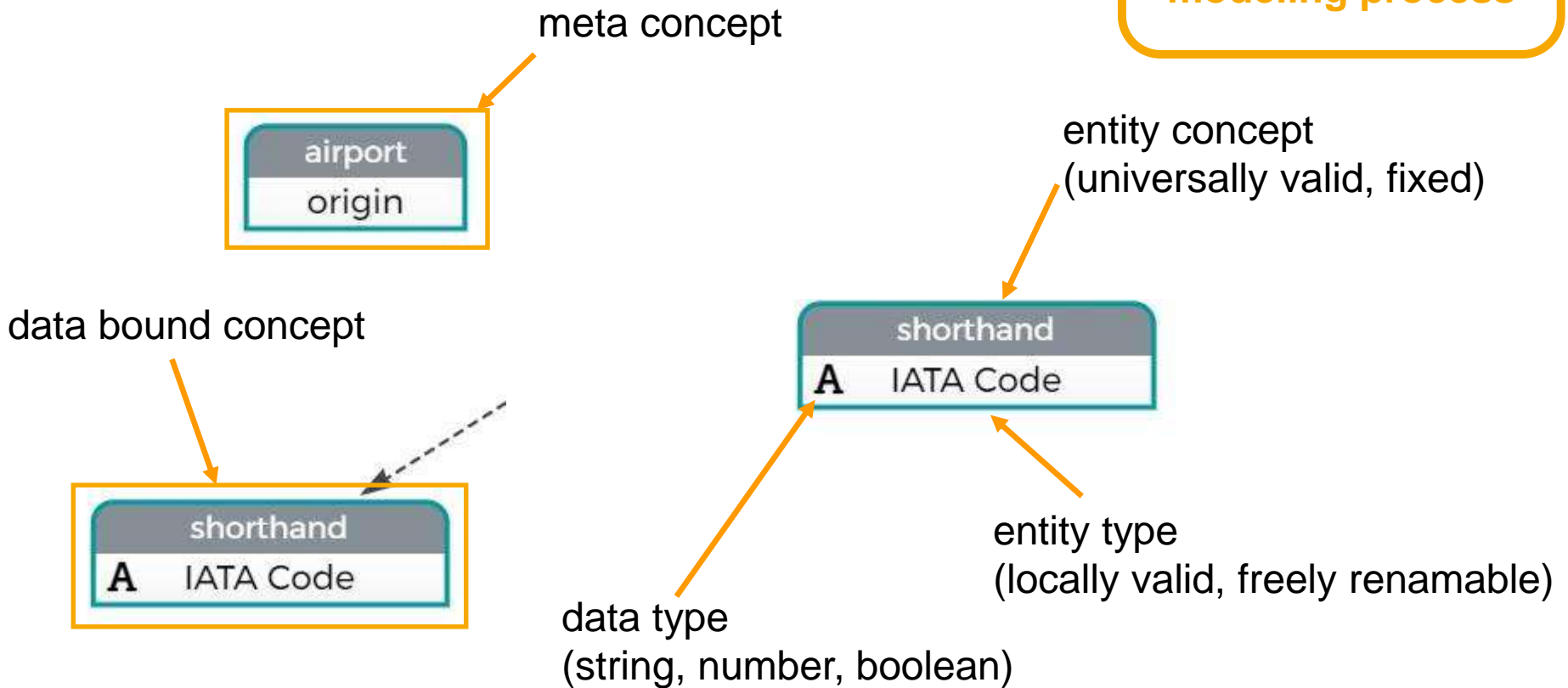


Use drag and drop as well known form of user input



Entity types can be freely modified

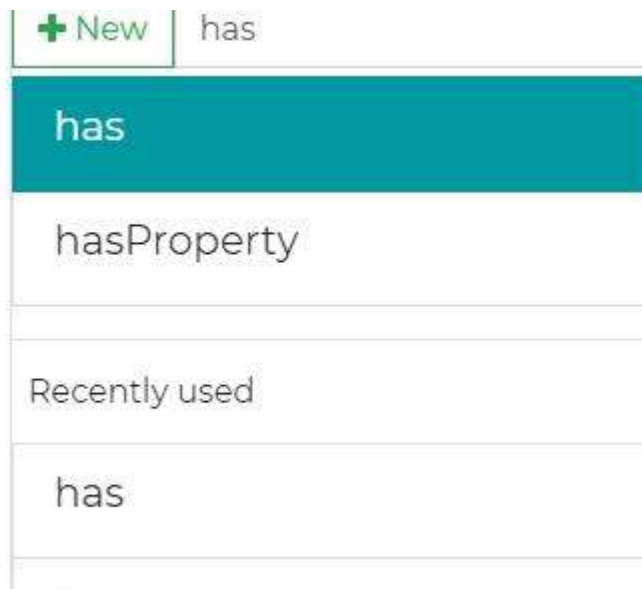
Do not restrict the user during the modeling process



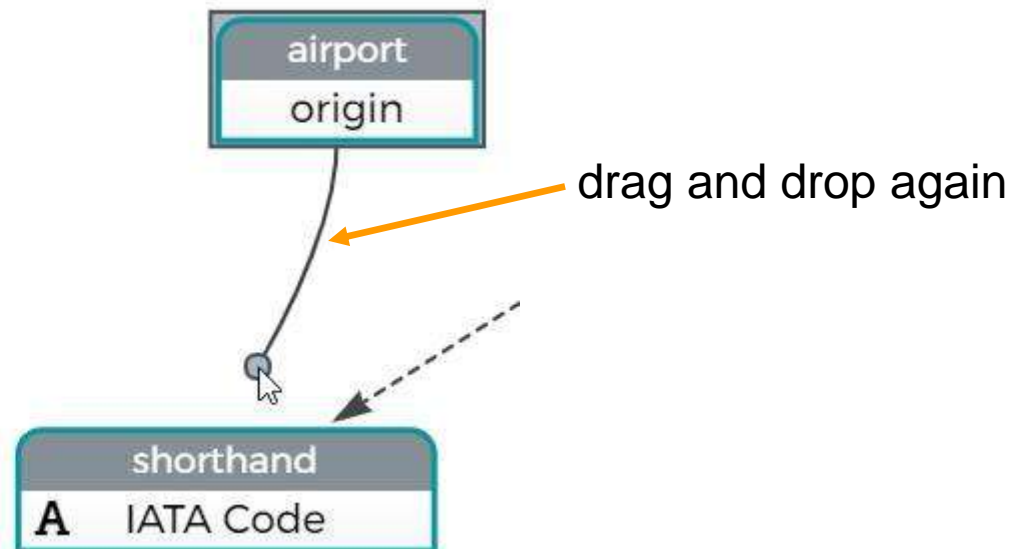
Handle relations the same way as entities

Do not restrict the user during the modeling process

Select ...

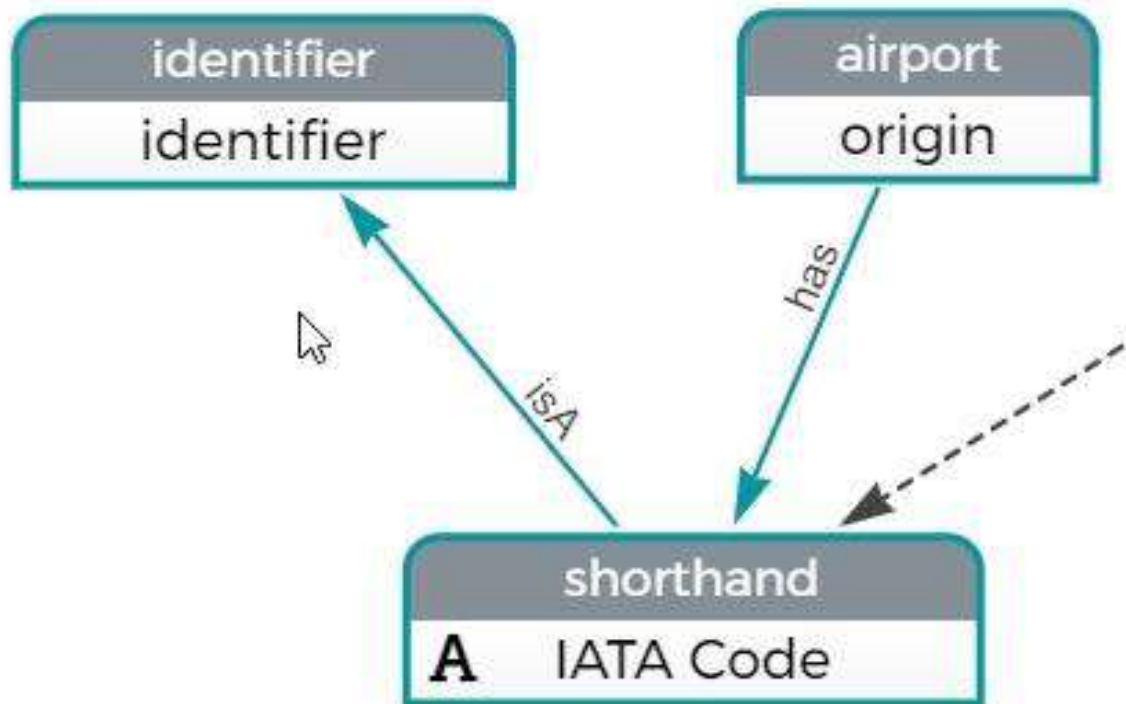


... and connect.



The more information the user gives, the stronger is the resulting model

Do not restrict the user during the modeling process



Modeling Interface

Modify the Data Schema

Model data values that contain more than one information

Adapt the schema to the user's needs

Define splitting patterns

Modify syntactic node

Edit splitting.

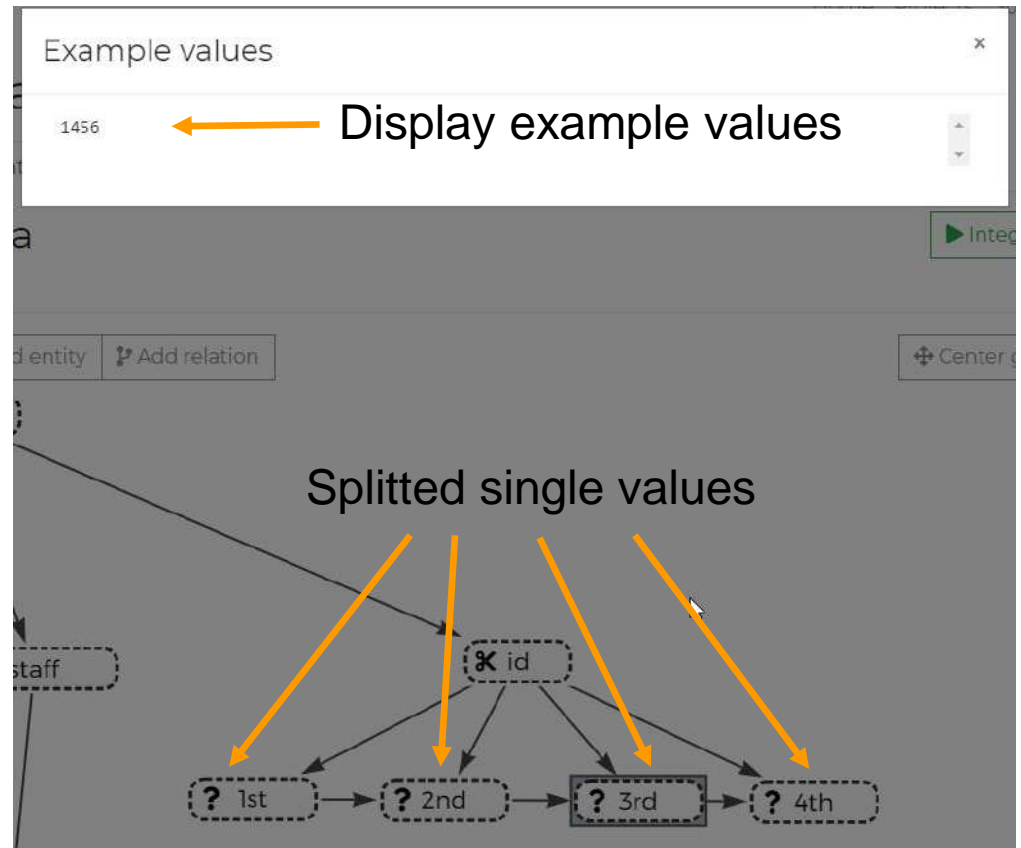
Modify the patterns for splitting a value. Each pattern must be a valid java.util.regex.Pattern literal. The pattern describe the location where the value is split NOT what should remain. Each pattern is applied exactly once to cut of one token from the start of the value. Thus n patterns split the value into n+1 tokens which become separate primitive nodes.

Split patterns

-
-
-

The list of patterns to split the value into multiple primitives. Each pattern must be a valid java.util.regex.Pattern literal. Each pattern is applied exactly once to cut of one token from the start of the value. Thus n patterns split the value into n+1 tokens which become separate primitive nodes.

Apply

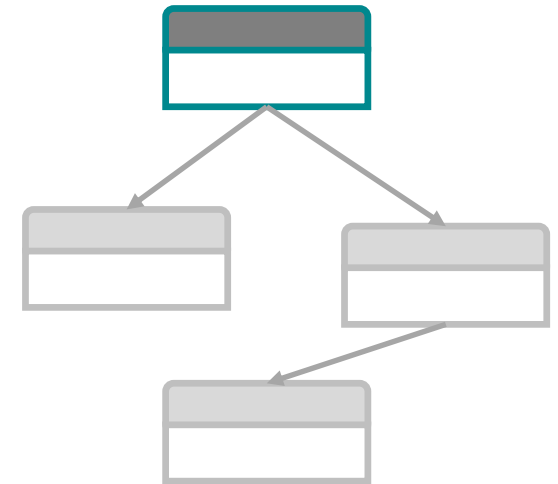


Modeling Interface Demonstration



Improving Handling and Support

- Improve zooming, focus on subtrees and auto alignment
- Change splitting dialog to be more user friendly
- Add multiple side projects to UI
 - Recommendation of Concepts and even subgraphs
 - Enable auto-detection of concepts
- Conduct greater user study
 - Compare to similar tools
 - Improve on modeling speed based on widespread user feedback



Thank you!

Alexander Paulus, M.Sc.

Scientific Researcher | Cognitive Computing & E-Health

Institute for Mechanical Engineering RWTH Aachen University

Tel: +49 241 / 80 91136

alexander.paulus@ima.rwth-aachen.de

Backup