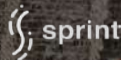


Sem4Tra Workshop

On the visualization of semantic-based mappings

Nicolò Pincioli, Mario Sacaj, Mersedeh Sadeghi, Safia Kalwar, Andreas Vogelsang and Matteo Rossi



September 6, 2021

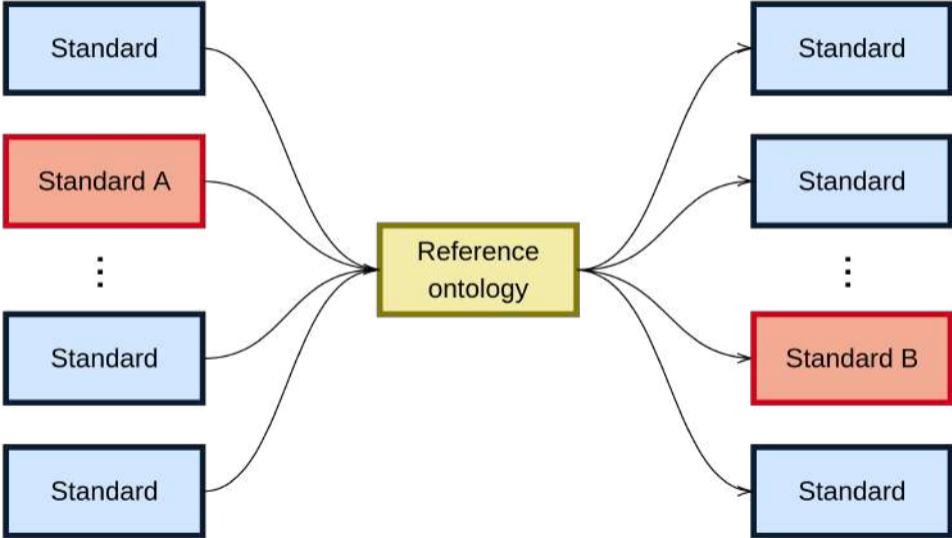


INFORMAÇÕES
Information

Introduction

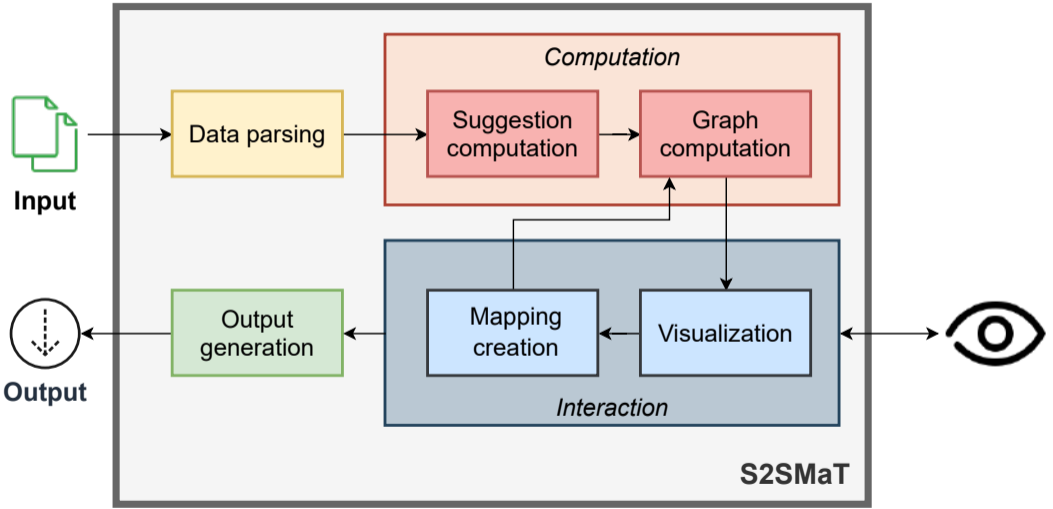


How can coordinated views be employed by users to translate terms from a standard to an ontology?



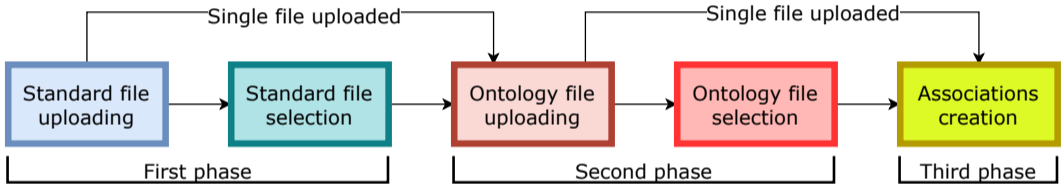
- Railway systems concepts represented using **ontologies**, hardly **readable** by humans;
- **Inaccurate** automatic associations creation;
- Effectiveness of **coordinated views**.

- **Ontologies** used only in specific sub-domains (e.g., railway safety [Hul+18] and the Dutch railways [BRS17]);
- Existing libraries (e.g., VOWL [Loh+16]) allow to represent **structured** data as graphs
- Existing tools allow to create **coordinated views**; [Vag+20], [Spu+20];
- Existing predictive **model** for suggesting associations.





User Interface Design



- Complex Type (XSD) \longleftrightarrow Class (Ontology)
- Element/Attribute (XSD) \longleftrightarrow Property (Ontology)

- Suggested one-to-one **mapping** between the concepts in **XSD** and the ones in the **ontology**
- Exploits a Word2vec-trained model (based on the Google News dataset)
- **Suggestions** → topmost similar terms
- Filtering based on binding representations
- Output → XSD-Ontology pairs

The image displays two software interfaces for ontology management. The left window, titled 'XSO Standard', shows a list of classes with their properties and types. The right window, titled 'Ontology', shows a graph visualization of these classes and their relationships.

XSO Standard Class Definitions:

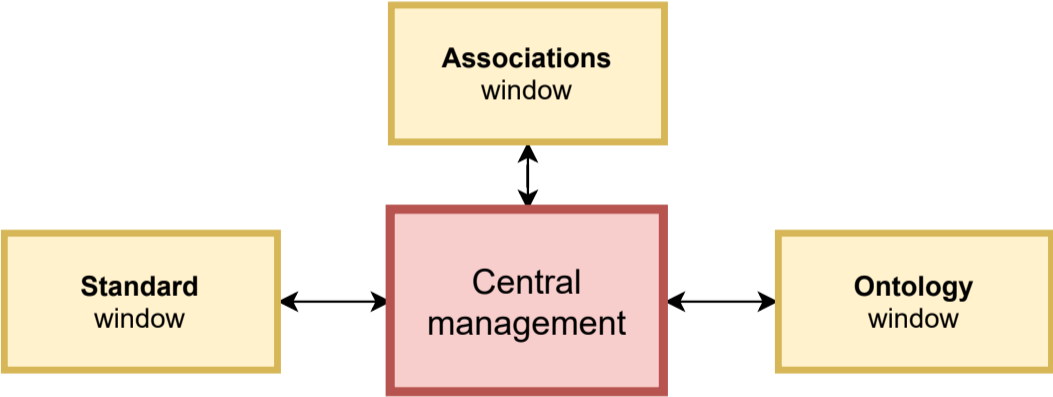
- PassengerTyped**
 - abstract
 - iri:foaf:01
 - iri:foaf:02/foaf:name
 - type:foaf:01/foaf:Name
 - type:foaf:01/foaf:Name
- DateOfBirth**
 - abstract
 - iri:foaf:01
 - iri:foaf:02/foaf:age
 - type:foaf:01
- Age**
 - abstract
 - iri:foaf:01
 - iri:foaf:02/foaf:age
 - type:foaf:01/foaf:age
- Identity**
 - abstract
 - iri:foaf:01
 - iri:foaf:02/foaf:age
 - type:foaf:01/foaf:age

Ontology Graph:

- Classes: `foaf:Person`, `foaf:PersonTyped`, `foaf:DateOfBirth`, `foaf:Age`, `foaf:Identity`.
- Relationships: `foaf:PersonTyped` is a subclass of `foaf:Person`. `foaf:DateOfBirth` and `foaf:Age` are subclasses of `foaf:PersonTyped`. `foaf:Identity` is a subclass of `foaf:PersonTyped`.

The 'Associations' window shows a table of associations between classes. It includes buttons for 'Add association' and 'Send data'.

Source	Target	Direction	Cardinality	Property
PassengerData	foaf:Person	→	1	foaf:name
DateOfBirth	foaf:DateOfBirth	→	1	foaf:age
Name	foaf:Person	→	1	foaf:name
LoyaltyCard	foaf:PersonTyped	→	1	foaf:age

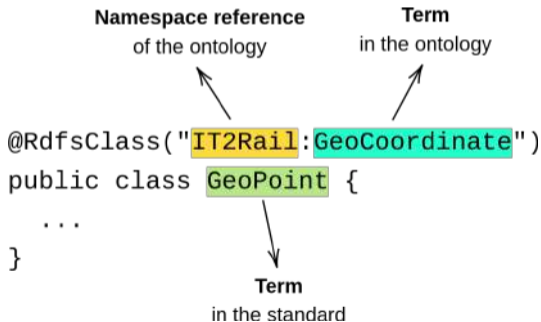


The image shows a screenshot of the WebVOWL interface, which is used for visualizing and editing ontologies. The interface is divided into several main sections:

- Left Panel (Axioms):** A list of properties and their domains. For example, 'Gender' is associated with 'Person', 'Address1' with 'Address', and 'City' with 'City'. A search bar at the top of this panel is labeled with a circled 'C'. A toolbar on the far left contains icons for zooming and navigation, with a circled 'B' next to it.
- Center Panel (Visualization):** A graph visualization of the ontology. Nodes represent classes and individuals, connected by lines representing relationships. A circled 'A' is placed at the top of this panel, and another circled 'A' is placed at the top of the right panel. A circled 'D' is placed over a 'Substitutions' dialog box at the bottom center, which has 'Association' and 'Send data' buttons. A circled 'G' is placed over a 'Standard' vs 'Ontology' selector at the bottom center. A circled 'F' is placed over a search and filter area at the bottom right.
- Right Panel (Details):** A detailed view of a selected class, 'Person'. It shows metadata such as 'Name', 'Type', 'Characteristics', and 'Description'. A circled 'A' is placed at the top of this panel. A circled 'E' is placed at the bottom right of the interface, near the 'Export' and 'Filter' buttons.

The annotations (A, B, C, D, E, F, G) highlight specific UI elements and their interactions, illustrating how Gestalt principles of grouping and proximity are used to organize the complex information presented in the interface.

- 1 XSD representation → Java constructs (mappings **materialization**)
 - Complex types → Java classes
 - Elements and attributes → Attributes and setter/getter methods
- 2 Annotation of Java constructs
- 3 Generation of a compressed file



- **Modular** application
- **Focus** on more intuitive data visualization and visualization principles
- **Integration** with the back-end
- Part of a **broader** project



Thank you for your attention

- [BOH11] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. “ D^3 Data-Driven Documents”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (Dec. 2011), pp. 2301–2309. ISSN: 1077-2626. DOI: 10.1109/TVCG.2011.185. URL: <https://doi.org/10.1109/TVCG.2011.185>.
- [BRS17] Bas Bach, Mark von Rosing, and Henrik von Scheel. “Using Ontology and Modelling Concepts to Develop Smart Applications: Example Dutch Railway”. In: *Int. J. Concept. Struct. Smart Appl.* 5.1 (2017), pp. 48–69. DOI: 10.4018/IJCSSA.2017010103. URL: <https://doi.org/10.4018/IJCSSA.2017010103>.
- [Cow01] Nelson Cowan. “The magical number 4 in short-term memory: A reconsideration of mental storage capacity”. In: *Behavioral and Brain Sciences* 24.1 (2001), pp. 87–114. DOI: 10.1017/S0140525X01003922.

- [Hul+18] Bernhard Hulin et al. “Towards a Common Ontology of Safety Risk Concepts for Railway Vehicles and Signaling”. In: *Computer Safety, Reliability, and Security - 37th International Conference, SAFECOMP 2018, Västerås, Sweden, September 19-21, 2018, Proceedings*. Ed. by Barbara Gallina, Amund Skavhaug, and Friedemann Bitsch. Vol. 11093. Lecture Notes in Computer Science. Springer, 2018, pp. 297–310. DOI: 10.1007/978-3-319-99130-6_20. URL: https://doi.org/10.1007/978-3-319-99130-6%5C_20.
- [Loh+16] Steffen Lohmann et al. “Visualizing Ontologies with VOWL”. In: *Semantic Web 7.4* (2016), pp. 399–419. DOI: 10.3233/SW-150200. URL: <http://dx.doi.org/10.3233/SW-150200>.
- [Mil94] George A Miller. “The magical number seven, plus or minus two: Some limits on our capacity for processing information.”. In: *Psychological review* 101.2 (1994), p. 343.

- [Spu+20] Maxim Spur et al. “Exploring Multiple and Coordinated Views for Multilayered Geospatial Data in Virtual Reality”. In: *Inf.* 11.9 (2020), p. 425. DOI: 10.3390/info11090425. URL: <https://doi.org/10.3390/info11090425>.
- [Vag+20] Nicolo Oreste Pinciroli Vago et al. “INTEGRA: An Open Tool To Support Graph-Based Change Pattern Analyses In Simulated Football Matches”. In: *Proceedings of the 34th International ECMS Conference on Modelling and Simulation, ECMS 2020, Wildau, Germany, June 9-12, 2020*. Ed. by Mike Steglich et al. European Council for Modeling and Simulation, 2020, pp. 228–234. DOI: 10.7148/2020-0228. URL: <https://doi.org/10.7148/2020-0228>.

A photograph of a modern subway station. The walls are curved and made of orange-colored panels. The ceiling is white with recessed lighting. A train with orange and blue panels is stopped at the platform. The floor is light-colored with a yellow safety line. A dark grey rectangular box is overlaid in the center of the image, containing the text "Visualization principles" in white.

Visualization principles

The screenshot displays the WebVOWL interface for an ontology. On the left, a tree view shows the ontology structure, with a search bar and a list of properties including Gender, LastName, FirstName, Address1-4, City, ZipCode, Country, MobilePhoneNumber, LandlinePhoneNumber, and EmailAddress. A search bar is also present at the top left. The main area shows a graph visualization of the ontology, with nodes representing classes and properties, and edges representing relationships. A search bar is located at the top center. On the right, a sidebar provides details for the selected class, including its name, type, and description. A search bar is also present at the top right. At the bottom, an 'Associations' panel shows a table of associations between classes, with a search bar and a 'Send data' button. The interface includes various navigation and search tools, such as a search bar at the bottom and a 'Send data' button.

A (top center)

A (top right)

B (left sidebar)

A (bottom center)

The image shows a screenshot of the WebVOWL interface, a tool for visualizing ontologies. The interface is divided into several main sections:

- Left Panel (Axioms):** Displays a list of axioms for the ontology. A search bar is at the top. A list of axioms includes: `Gender`, `LastName`, `FirstName`, `Address1`, `Address2`, `Address3`, `Address4`, `City`, `ZipCode`, `Country`, `MobilePhoneNumber`, `LandlinePhoneNumber`, and `EmailAddress`. A `complexContent` section is also visible, containing a `base="common:FSM.ID"` axiom. A search bar is located above this list.
- Center Panel (Ontology Graph):** A central visualization of the ontology graph. Nodes are represented by circles, and relationships by lines. The graph shows a central node `Person` connected to various other nodes like `Gender`, `Address`, `City`, etc. The graph is rendered in a light blue and green color scheme.
- Right Panel (Information Panel):** Provides metadata and statistics for the selected ontology. It includes fields for Name, Type, Characteristic, Description, Metadata, and Selection Details. The description for the selected ontology is: "Description: Description of roles associated with a Fare Product Description by a lead description - possibly multilingual - of Fare Rules. Typically used in the public transport system in the webvowl project. Date: 2016-06-01. Creator: J. Lehmann. Contributor: J. Lehmann. WebvowlInfo: 2016-06-01. Created as part of event: European 2016-06-01 (changed to FareRuleDescription)".
- Bottom Panel (Navigation and Tools):** Contains a search bar, a list of tabs (Standard, Ontology), and various navigation icons. A search bar is also present at the bottom right.

Annotations A through G are placed on the screenshot to highlight specific UI elements:

- A:** Points to the search bar in the top left and the search bar in the top right.
- B:** Points to the navigation icons on the left side.
- C:** Points to the central ontology graph.
- D:** Points to the tabs (Standard, Ontology) at the bottom.
- E:** Points to the search bar at the bottom right.
- F:** Points to the navigation icons at the bottom right.
- G:** Points to the search bar in the bottom left.

Short memory holds a limited amount of items [Mil94; Cow01]

- 1 Shallow menus
- 2 Visual feedback after selection
- 3 Standard and consistent icons

Thinking is interactive

- ① Windows can be moved, resized and set always on top
- ② Graphs can be dragged, zoomed, coloured, ...
- ③ Variable amount of details

The screenshot displays a software interface for ontology management, likely Protégé. The main window shows a central ontology graph with 'owl:Thing' at the center, connected to various classes and properties. A search bar at the top right contains 'LoyaltyCard'. An 'Associations' window is open in the foreground, showing a table of associations between 'LoyaltyCard' and 'TCDataPKG:hasLoyaltyCardType'. The table has columns for 'Standard' and 'Ontology'. The 'Standard' column contains 'LoyaltyCard' and the 'Ontology' column contains 'TCDataPKG:hasLoyaltyCardType'. The 'Associations' window also features buttons for 'Add association' and 'Serial view'. On the left side, a sidebar lists ontology elements: 'Voucher', 'LoyaltyCard', 'Name', 'FirstName', 'Gender', and 'PassengerRefsList'. On the right side, a panel titled 'IT2Rail Ontology' provides metadata and statistics for the selected ontology.

Standard	Ontology
LoyaltyCard	TCDataPKG:hasLoyaltyCardType

The image shows a screenshot of a software application interface for ontology management, divided into several panels:

- Left Panel (OWL2 Standard):** Displays a tree view of classes and properties. The classes listed are `Voucher`, `LoyaltyCard`, `Name`, `FirstName`, `Gender`, and `PassengerRefsList`.
- Top Center Panel (Ontology):** Shows a central node `owl:Thing` with a dense network of outgoing relationships to various other classes and properties. A search bar is visible above the network.
- Right Panel (IT2Rail Ontology):** Provides detailed metadata for the selected ontology. It includes fields for `Version`, `Author`, `Language`, `Description`, `Metadata`, `Statistics`, and `Selection Details`.
- Bottom Center Panel (Associations):** A dialog box for creating or editing associations. It features a table with columns for `Standard` and `Ontology`. The table contains one entry: `LoyaltyCard` (Standard) associated with `TCDataPKG:hasLoyaltyCardType` (Ontology).

The image displays two side-by-side screenshots of OWL editors. The left screenshot shows a tree view of an ontology with text labels of various sizes. The right screenshot shows a radial graph of the same ontology, where text labels are also of various sizes. A small dialog box titled 'Associations' is overlaid on the right screenshot, showing a table of associations with columns for 'Standard' and 'Ontology'.

Associations

Standard	Ontology
LoyaltyCard	TCCDataPKG:hasLoyaltyCardType

The image displays two screenshots of ontology editors side-by-side, illustrating the impact of text size on readability.

Left Screenshot (Prolog Standard): Shows a hierarchical tree structure of an ontology. The root node is "owl:Thing". Major branches include "PassengerData", "PassengerPreference", and "DiningPassengerPreference". Under "PassengerData", there are sub-classes like "ComplexContext" and "EmailCommonPCIM ID". The text is large and clear, making the structure easy to navigate.

Right Screenshot (Orkney): Shows a dense network graph of the same ontology. The central node is "owl:Thing". Numerous property names radiate from it, such as "mob.hasMileage", "mob.hasRefToFull", "TCDDataPKG.has...", and "customer.hasRef...". The text is significantly smaller than in the left screenshot, making it difficult to read the specific property names and their relationships.

Both screenshots include a search bar at the top right and a sidebar on the left with navigation icons. The right screenshot also features a detailed sidebar on the right with metadata and statistics for the selected class.

The image displays two side-by-side screenshots of ontology editing software. The left window, titled "RDF Standard", shows a hierarchical tree structure of classes. The root node is "owl:Thing", which branches into several categories, including "Passenger Data", "Passenger Preferences", and "Shopping Passenger Preferences". Each category contains a list of specific classes, such as "PassengerData", "PassengerPreferences", and "ShoppingPassengerPreferences". The right window, titled "Orkney", shows a radial graph of classes centered on "owl:Thing". The graph displays numerous classes and their relationships, with some classes highlighted in red. A search bar is visible at the top of the Orkney window. Below the graph, there is a sidebar for "IT2Rail Ontology" with various tabs and a search bar. An "Applications" dialog box is open in the foreground, showing "Add association" and "Send data" buttons, and a table with columns "Standard" and "Orkney".

The image displays two side-by-side screenshots of OWL editors. The left window, titled 'RDF Standard', shows a graph-based ontology editor with a search bar and a list of properties on the right. The right window, titled 'Orkney', shows a dense, circular graph of classes and properties, with a central node labeled 'owl:Thing'. A detailed sidebar on the right of the 'Orkney' window provides metadata for the 'IT2Rail Ontology', including version, author, language, and description. An 'Associations' dialog box is also visible in the foreground of the 'Orkney' window.

IT2Rail Ontology
Version: 1.0
Author: Robert
Language: english
Description: IT2Rail Ontology for...
Metadata
Date: 2014-05-21 10:00:00
System: JLeveson
Language: english
Statistics
Selection Details
Name: owl:Thing
Type: owl:Class
Description: The person performing work on an...
Language: english
URI: http://www.it2rail.org/ontology/IT2RailOntology.ttl
Version: 2014-05-21 10:00:00
Author: Robert
Language: english

Associations
Add association Send data
Standard Orkney
LennyCard TCDataPKG:has...
owl:Thing

The image displays a complex software interface for ontology management, divided into several key sections:

- Left Panel (RDF Standard):** Shows a hierarchical tree structure of classes. The root node is `Person` (labeled "Person (class)"), which branches into `Reference` and `ContactInformation`. `Reference` further branches into `ReferenceType` and `ReferenceValue`. `ContactInformation` branches into `contactContent` and `Person (class)`. The `Person (class)` node has a large number of associated properties listed on the right, including `Gender`, `Lastname`, `Firstname`, `Address1`, `Address2`, `Address3`, `Address4`, `City`, `ZipCode`, `Country`, `MetaPhoneNumber`, and `LastPhoneContact`.
- Center Panel (Ontology):** A dense network graph with a central node `owl:Thing`. Numerous other nodes are connected to it, representing various classes and properties from different ontologies, such as `TCDataPKG:has...`, `customer:hasRef...`, `mob:hasMileage`, and `product:hasRefTo...`.
- Right Panel (IT2Rail Ontology):** A detailed view of the `IT2Rail Ontology`. It includes a search bar, version information, a description, and metadata. The `Statistics` section shows the number of classes and instances. The `Selection Details` section provides information about the selected class, including its name, type, and description.
- Bottom Panel (Applications):** A window titled "Applications" with buttons for "Add association" and "Send data". It lists several applications, including "LennyCard", "TCDataPKG:has...", and "mob:hasMileage".

The screenshot displays two ontology editors side-by-side. The left editor shows a graph with nodes like 'PassengerData', 'complexContent', and 'base'. The right editor shows the 'IT2Rail Ontology' with a central 'mob:Passenger' node highlighted in red. An 'Associations' dialog box is open, showing a table of associations between 'PassengerData' and 'mob:Passenger'.

Standard	Ontology
PassengerData	mob:Passenger
DateOfBirth	mob:hasDateOfBirth
Name	mob:hasPassengerName

Association creation

The screenshot displays the Protege ontology editor interface. The central 'Associations' window is the primary focus, showing a table with the following data:

Standard	Classing
PassengerData	mob:Passenger
DateOfBirth	mob:hasDateOfBirth
Name	mob:hasPassengerName
LoyaltyCard	Infrastruc...

The background ontology graph features a central node 'owl:Thing' with numerous outgoing edges to various property instances, such as 'TCDDataPKG:has...', 'interop:hasFe...', 'mob:isVa...', 'mob:hasProvide...', 'shopping:h...', 'mob:hasTra...', 'interop:hasR...', 'BookingS...', 'ProvidedBy', 'customer:hasF...', 'product:hasRefTo...', 'mob:hasRefToEnt...', 'mob:hasMe...', 'product:has...', 'infrastructure#h...', 'mob:hasMobilityR...', and 'product:hasComm...'. The right sidebar provides details for the 'LoyaltyCard' class, including its description: 'The person purchasing travel on an identity or a travel card on board a vehicle' and its metadata.

The screenshot displays the Protege ontology editor interface. On the left, a class hierarchy is visible, including 'Voucher', 'LoyaltyCard', and 'PassengerRefsList'. The main workspace shows a central 'owl:Thing' node with numerous outgoing property links. A dialog box titled 'Associations' is open, showing a table with columns for 'Instances' and 'Domain'. The table contains the following rows:

Instances	Domain
PassengerData	mob:Passenger
DateOfBirth	mob:hasDateOfBirth
Name	mob:hasPassengerName

Below the table, there are fields for 'LoyaltyCard' and 'TCDataPKG:hasLoyaltyCardType'. A red box highlights the text 'TCDataPKG:has...' in the background ontology view.

Implementation



- Integration and adaptation of existing visualization libraries and tools [Loh+16; BOH11; Vag+20]
- Implementation of a modular web-based application
- Integration with the back-end system

- Each module creates independent visualizations
- One module corresponds to one window
- User interactions are managed on the client side

- Data are requested using JavaScript requests
- Asynchronous requests don't block the interface while waiting
- Each module performs independent requests



Input and output examples

```
<xsd:complexType name="Identity">
  <xsd:complexContent>
    <xsd:extension base="common:FSM.ID">
      <xsd:attribute name="IdentityCard" type="xsd:string" use="required">
        </xsd:attribute>
      ...
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="PassengerPreference">
  <xsd:sequence minOccurs="0" maxOccurs="1">
    <xsd:element name="SpecialOrder" type="tariff:SpecialOrder" minOccurs
      ="0" maxOccurs="unbounded">
    </xsd:element>
    <xsd:element name="Preference" type="passenger:Preference" minOccurs="0"
      maxOccurs="unbounded">
    </xsd:element>
    ...
  </xsd:sequence>
</xsd:complexType>
```

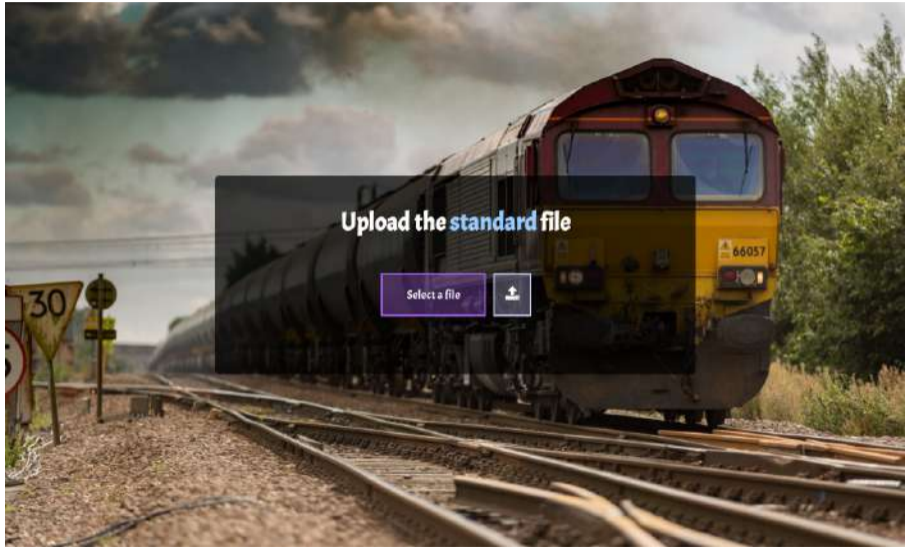
```
<owl:Class rdf:about="http://www.it2rail.eu/ontology/InteroperabilityFramework#
  PrivateStation">
  <rdfs:subClassOf rdf:resource="http://www.it2rail.eu/ontology/
    InteroperabilityFramework#StopPlace"/>
  <dc:contributor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Robert Lehmann</dc:contributor>
  <dc:creator rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Robert
    Lehmann</dc:creator>
  <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime
    ">2016-07-12T10:57:53Z</dc:date>
  <dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"/>
  <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    2016-07-12 initial version 2017-08-30 moved to namespace
      InteroperabilityFramework
  </owl:versionInfo>
</owl:Class>
```

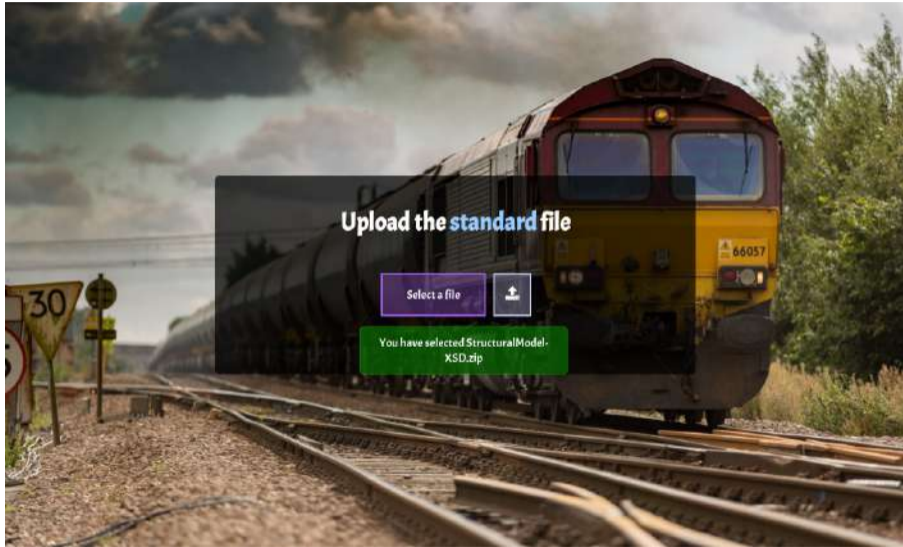
```
<owl:ObjectProperty rdf:about="http://www.it2rail.eu/ontology/transport#
  hasTransportServiceProviderID">
  <rdfs:subPropertyOf rdf:resource="http://www.it2rail.eu/ontology/hasID"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  ...
  <dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"/>
  <an:i2rumlDomain rdf:datatype="http://www.w3.org/2001/XMLSchema#string
    ">TransportServiceProvider</an:i2rumlDomain>
  <an:i2rumlRange rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    TransportServiceProviderID</an:i2rumlRange>
  <owl:deprecated rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">
    true</owl:deprecated>
  <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string
    ">2017-02-08 initial version</owl:versionInfo>
</owl:ObjectProperty>
```

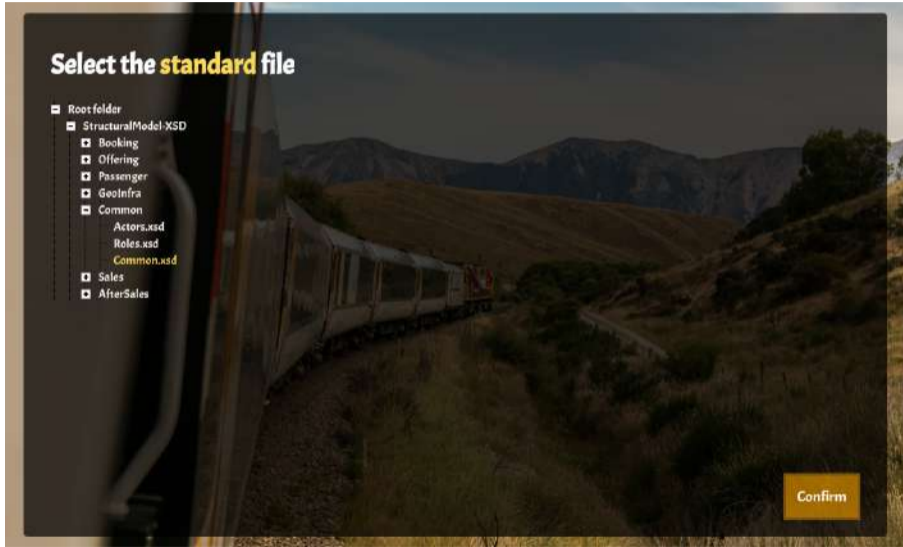
```
@RdfsClass("mob:Passenger")
@NameSpaces(...)
public class PassengerData extends FSMID
{
    @XmlElement(name = "DateOfBirth")
    @XmlSchemaType(name = "date")
    @RdfProperty(propertyName = "mob:hasDateOfBirth")
    protected XMLGregorianCalendar dateOfBirth;
    @XmlElement(name = "Voucher")
    protected List<Voucher> voucher;
    ...
    public List<String> getIdentityTypeId() {
        if (identityTypeId == null) {
            identityTypeId = new ArrayList<String>();
        }
        return this.identityTypeId;
    }
}
```


Workflow details









- Syntactic validity
- File parsing
- Pre-processing
- Cleaning

- Suggested one-to-one **mapping** between the concepts in **XSD** and the ones in the **ontology**
- Exploits a Word2vec-trained model (based on the Google News dataset)
 - Word \rightarrow 300-dimensional feature vector
 - Relative distances between vectors \longleftrightarrow semantic similarity
- **Suggestions** \rightarrow topmost similar terms
- Filtering based on binding representations
- Output \rightarrow XSD-Ontology pairs