Appendix

Capabilities of the Semantic Treehouse[1] vocabulary hub:

1. Version management of ontologies – assume an industry specific ontology changes along innovations in the field. The way Semantic Treehouse assists in such a situation is by offering the user the capability to define the new version and change the ontology accordingly. We give an example of versioning of an ontology, ReplanIT, in Figure 1.

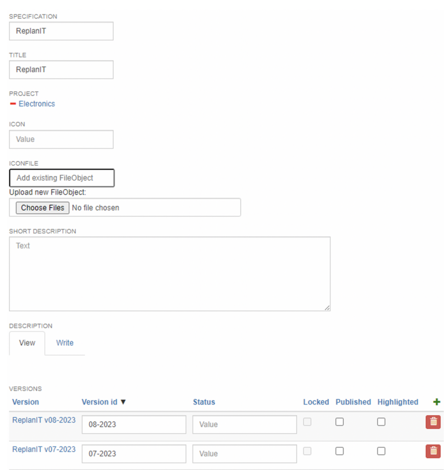

Fig. 1. Semantic Treehouse: Version Management of Ontologies

2. Ontology interoperability when designing an interaction – assume a user wants to create an interaction with a business partner based on multiple ontologies from different industries. In Semantic Treehouse the user may include as many ontologies as they want, as highlighted in Figure 2.
3. Customizable interactions – assume a user want to create an interaction with a business partner based on multiple ontologies from different industries. In Semantic Treehouse, the user is prompted with the decision of which attributes to include in the interaction. Since the user might not be interested in specifying all the possible attributes, having this feature saves from namespace pollution.
4. Mapping elements of an interaction to a standard/regulation. In Semantic Treehouse the user is allowed to import standards or regulations as a taxonomy.

Figure 3 is an example of mapping where in an interaction based on an EV battery, the attribute Battery IDDM Code addresses Article 2(55a) (stating "a

[1] https://www.semantic-treehouse.nl

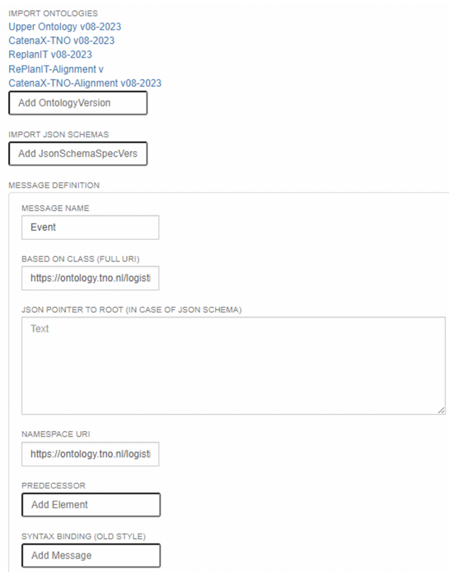

Fig. 2. Semantic Treehouse: Multiple Ontologies used for defining an Interaction

unique string of characters for the identification of batteries that also enables a web link to the battery passport") of the Battery Pass.

An overview of the solution is presented on Figure 4 and we will describe its' elements top to bottom. First, in the Upper Ontology section we denote the RDF-based upper level ontology. The following 2 sections represent the industry specific ontologies, the ReplanIT ontology for electronics and CatenaX-TNO for batteries. Additionally, for each industry, the chosen alignment is saved as a distinct ontology, to enable easier updates and lower maintenance complexity. In the last section we denote 3 message models, which represent an business interaction. The first message model has been built to highlight the interoperability of the electronics and upper ontology via the alignment.

When constructing the custom interaction, the user of Semantic Treehouse can include attributes that are part of the electronics ontology (Figure 6, the attributes inside the orange rectangle) or attributes part of the upper ontology (Figure 6, attributes inside purple rectangle). The second message model highlights the interoperability of all 3 ontologies where an event includes a product described using attributes from the electronics ontology as well as a product described using attributes from the battery ontology. Finally, the third message model highlights the interoperability of the upper ontology and the battery ontology.

Reflecting on the knowledge requirements to achieve an interoperable solution between industries (as depicted in Figure 18a earlier in this section) we highlight how these needs are addressed when using Semantic Treehouse. First,

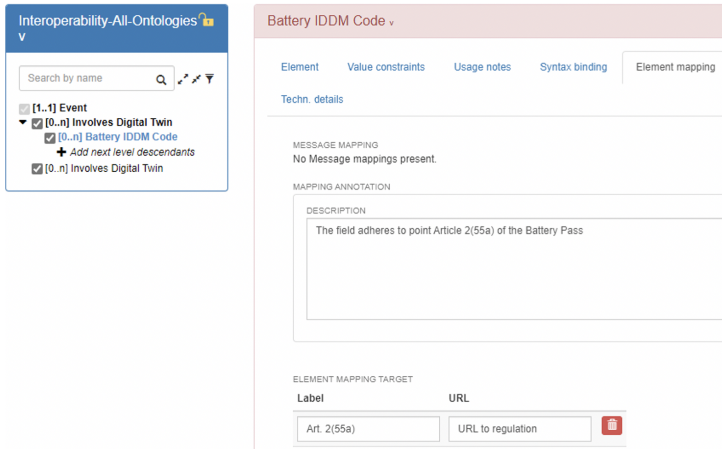

Fig. 3. Element mapping: An Attribute of an Interaction mapped to a Standard/Regulation

industry specific ontologies may require updates as new innovations are introduced, Semantic Treehouse offers the possibility of versioning ontologies and also has semantic mapping capabilities for a smoother transition of legacy systems. Second, alignments may introduce subjectivity because of the required human intervention, Semantic Treehouse offers public and federated access capabilities, so partnering stakeholders can review and discuss or possibly change the alignments. Finally, Semantic Treehouse also offers a user-friendly Graphical User Interface (GUI) that does not require extensive semantic knowledge, enabling users to create interactions (message models) without being semantic experts.

Figure 6 presents a practical example of interoperability between companies of different industries (batteries and automotive) would be a EV battery producer company may host their choice of domain specific ontology and its' updates (which could be hosted by the company itself or in open repositories on platforms like Semantic Treehouse), so their (car) OEM business partners may use for future transactions. Defining interactions for business transactions with the ontology of the EV battery producer would allow the (car) OEM to know what information can be queried, speeding up the business transaction process. The EV battery producer thus defines what information capabilities they have, while the(car) OEM partner conforms to this ontology when sending queries. For example, the (car) OEM could ask for the lithium content of the EV battery, based on the EV battery producer's ontology.

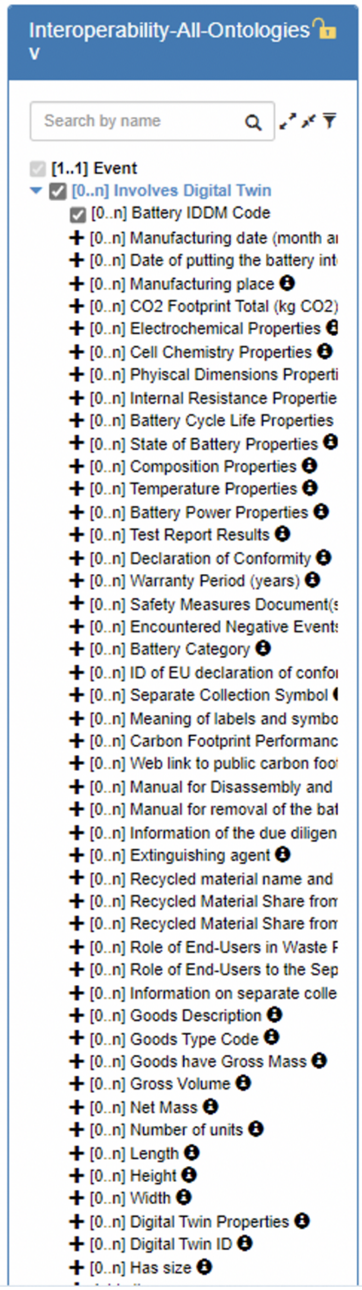

Fig. 4. Decision: GUI where the user may select attributes to include in the interaction

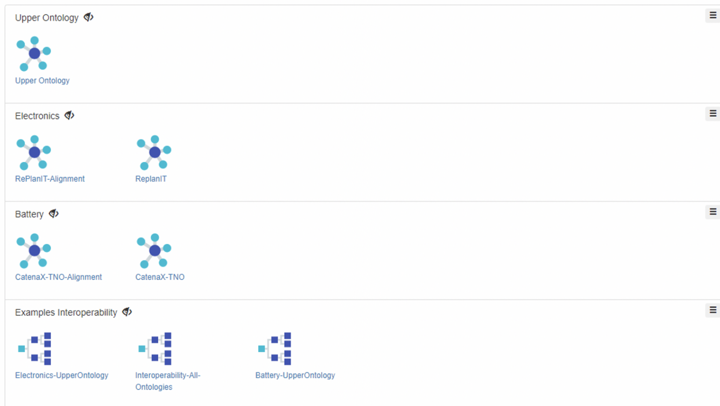

Fig. 5. Semantic Treehouse Overview

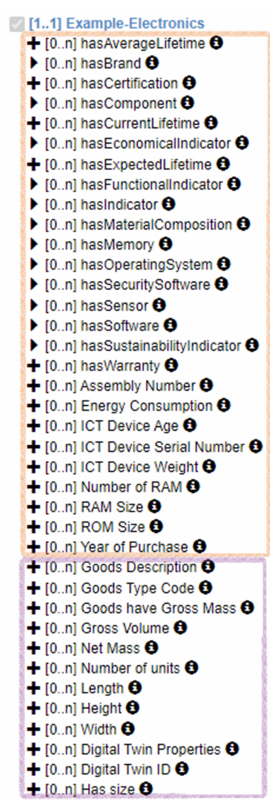

Fig. 6. Semantic Treehouse: Message Model Interoperability Electronics and Upper Ontology

We form the following triple statements to visualize an example (see below) and we provide 2 competency questions to highlight how a SPARQL query may be formed to request the information given the above triples (see Table 1).

```
 :plastic_casing a DigitalTwin:Product;
    DigitalTwin:digitalTwinID ‘‘PC-01’’;
        DigitalTwin:netMass 2;

 :glass_piece a DigitalTwin:Product;
      DigitalTwin:digitalTwinID ‘‘GP-01’’;
      DigitalTwin:netMass 10;

:control_system a replanit:ICTDevice;
     DigitalTwin:digitalTwinID ‘‘CS-01’’;
     DigitalTwin:netMass 0.2;
     replanit:ICTDeviceAge 0;
     replanit:ICTDeviceSerialNumber ‘‘f48a5929-3a49-4caf-941f-f8eedc61b5d7’’;
     replanit:hasBrand ‘‘NoBrand’’

 :switch a replanit:HardwareComponent;
     DigitalTwin:digitalTwinID ‘‘S-01’’;
     replanit:hasCurrentLifetime ‘‘2 years’’;
     replanit:hasExpectedLifetime ‘‘2.5 years’’;

Event:switch_control_system_Association a Event:Association;
Event:involvesDigitalTwin :control_system, :switch

:car_window a DigitalTwin:Product;
     DigitalTwin:digitalTwinID ‘‘CW-01’’;
     DigitalTwin:netMass 12.2;

Event:plastic_casing_car_window_Association a Event:Association ;
Event:involvesDigitalTwin :car_window, :plastic_casing
Event:glass_piece_car_window_Association a Event:Association ;
 Event:involvesDigitalTwin :car_window, :glass_piece
Event:control_system_car_window_Association a Event:Association ;
 Event:involvesDigitalTwin :car_window, :control_system
```

Table 1. Example of competency questions and SPARQL Query

Competency Questions	SPARQL Query

What is the expected lifetime of the switch of the car window with identifier "CW-01" from this moment?	SELECT (?hasExpectedLifetime - ?hasCurrentLifetime AS ?expectedLifetimeLeft)
	WHERE {
	?carWindow a DigitalTwin:Product .
	?carWindow DigitalTwin:DigitalTwinID "CW-01" .
	?associationEvent a Event:AssociationEvent .
	?associationEvent Event:involvesDigitalTwin ?carWindow, ?controlSystem .
	?associationEvent Event:involvesDigitalTwin ?controlSystem ?switch .
	?switch replanit:hasExpectedLifetime ?hasExpectedLifetime
	?switch replanit:hasCurrentLifetime ?hasCurrentLifetime
	}

What is the weight of the plastic casing of the car window with identifier "CW-01"?	SELECT ?weight WHERE {
	?carWindow a DigitalTwin:Product .
	?carWindow DigitalTwin:DigitalTwinID "CW-01" .
	?associationEvent a Event:AssociationEvent .
	?associationEvent Event:involvesDigitalTwin ?carWindow, ?plasticCasing .
	?plasticCasing DigitalTwin:netMass ?weight .
	}