

# › THE USE OF THE SEMANTIC WEB TECHNOLOGIES FOR PROVIDING PORTFOLIO-LEVEL END OF LIFE ANALYSIS FOR TRANSPORT INFRASTRUCTURE

## END OF LIFE RULE-CHECKING MODEL: THE CASE OF NAVIGATION LOCKS



Esra Bektaş, PhD

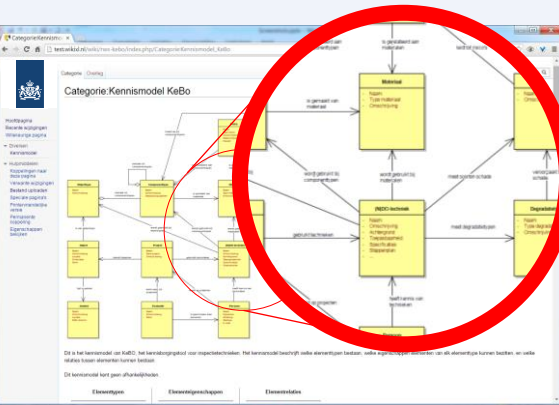


# Topics to cover

Background work	<ul style="list-style-type: none"><li>• Knowledge-bases for the Dutch Road Authority (Rijkswaterstaat)</li><li>• Semantic Media Wiki Implementations</li></ul>	2'
Research Project as a Use Case	<ul style="list-style-type: none"><li>• Sharing The Idea &amp; Exercising the Semantic Web Technologies</li><li>• via End of Life Rule Checking Model (<b>on-going project</b> - PoC)</li></ul>	10'
Next Steps	<ul style="list-style-type: none"><li>• + Main take aways, Upscale PoC To Locks &amp; Later New Object Types</li></ul>	3'

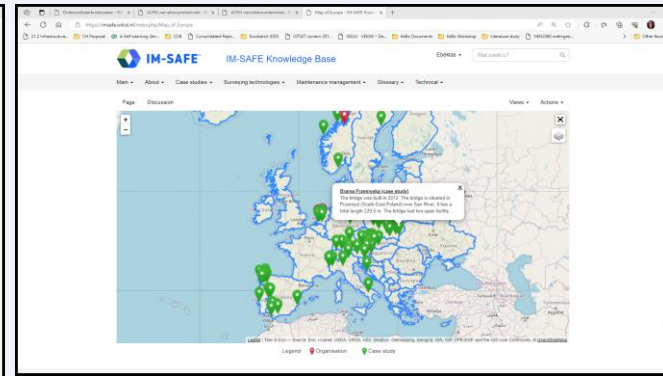
# Background Work

- Knowledge-Bases At the Dutch Road Authority
  - A Semantic Wiki Implementation, always an object type and associated knowledge needs
  - Pre-determined queries, Coordinated text, images, inspection data
- End-user oriented solution together with ArchiXL



The screenshot shows a search interface for the IM-SAFE Knowledge Base. It features a search bar and several filter options for refining search results:

- Home Kennisbron Onderzoekstechnieken Gebruik van deze wiki
- Onderzoekstechniekzoeker
- Selecteer een beheersobject: Brug (vrij)
- Selecteer een element: Rijk
- Selecteer materiaal: Staat
- Filter op materiaalomschrijvingen (optioneel): (Aant meerder materiaalomschrijvingen invoeren.)
- Filter op soort schade (optioneel): Gansse (Aant meerder soorten schade invoeren.)
- Filter op dagvrijdagte (optioneel): (Aant meerder dagvrijdagte invoeren.)
- Filter op status van de techniek (optioneel):  gebruik  experimenteel  niet meer in gebruik
- Start
- Her vinden van een geschilde Onderzoekstechniek. Kun eniger recorden in beeld brengen.





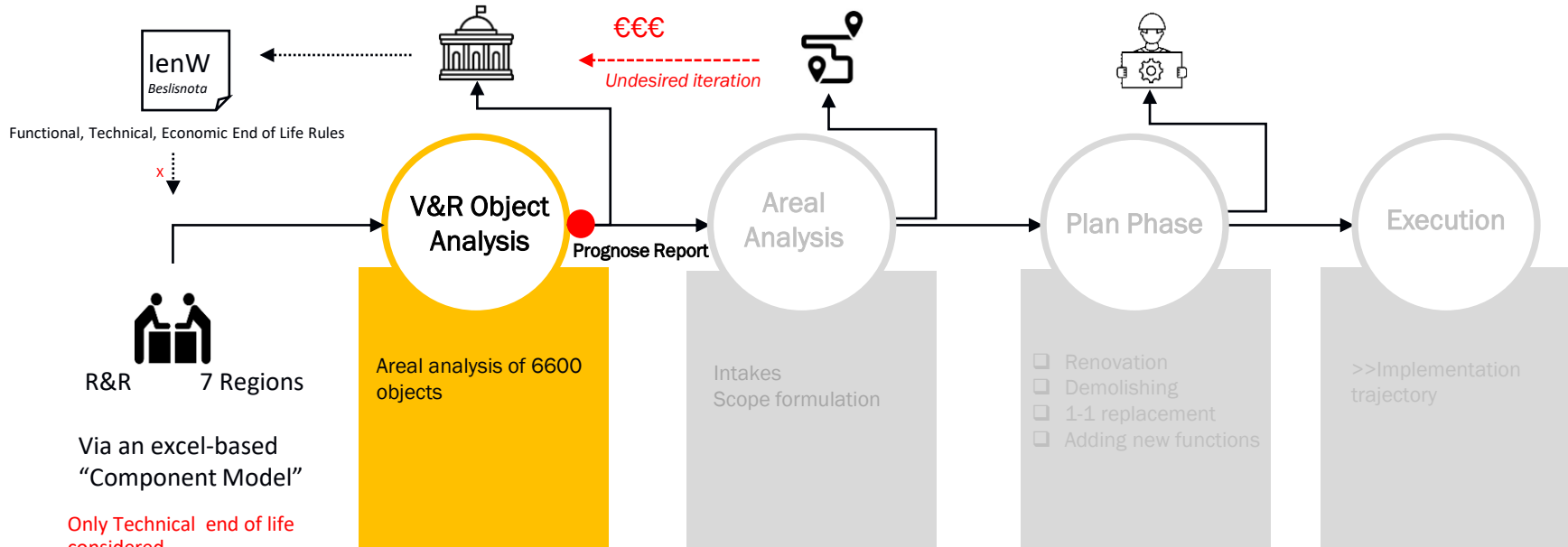
**TNO** innovation  
for life

# End of life rule-checking model for asset prioritization: Navigation Locks

As Knowledge Development Project

# Context: End of Life Decision Moment

Rapid analysis of the asset portfolio

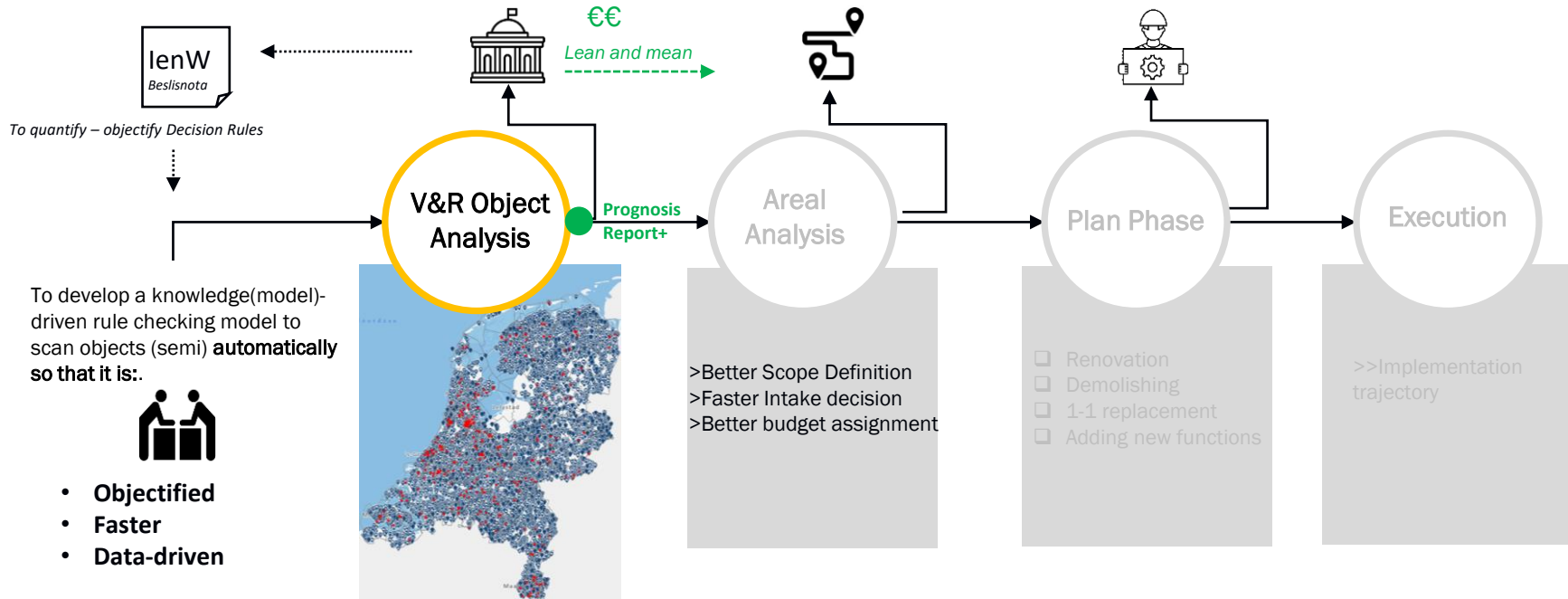


Via an excel-based  
"Component Model"

Only Technical end of life  
considered

Rule of thumb: concrete bridges <  
1976, replace 10%

# Approaching with the use of Semantic Web Technologies

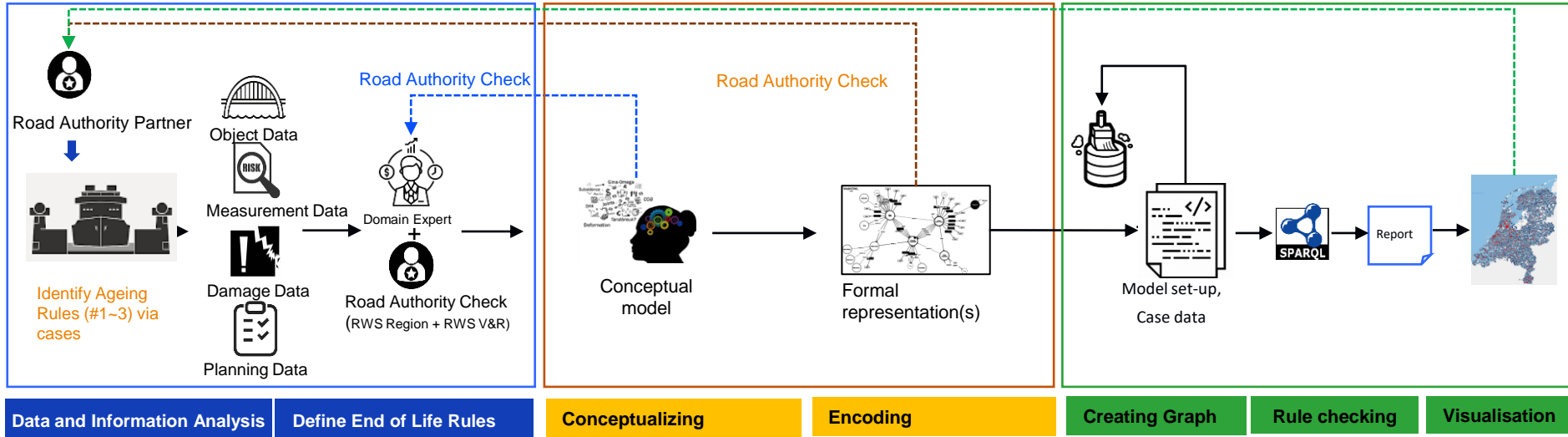


# Methodology, envisioned

## Data and Knowledge acquisition

## Formalization

## Development



# End of Life Rules

There are formal definitions Based on I&M (2013), VONK (2012) but also Klanker et. Al (2016), Bakker et al (2016), Wessels et al. (2018) They all interact on the level of object properties.

## Technical end of life

- Structural safety can no longer be guaranteed.
- Or when a structure has serious structural defects.
- Or when a (critical) component is obsolete.

## Functional end of life

- When an asset no longer fulfills its designed function due to changing environment, requirements, demand.
- When there are new functions needed from the asset itself or the location where the asset is.

## Economic end of life

- When an asset becomes too expensive to maintain.
- When EELI indicator of Bakker et al.(2016) comes closer to 1.

We published the findings in the IALCCE 2018.



# End of Life Rules for Navigation Locks

## Delden, Eefde, Hengelo Locks at the Twente Canal What makes navigation locks to be as V&R candidates?



**Situatie**      **Funcie**      **Niet - Beschikbaarheid**

Huidige situatie	Passeerfuncie	[uur/jaar]
		400,9
		122,7
		6,7
		388,3

**Herijking capaciteitanalyse Vaarwegen**  
Periode tot 2026 (Nota Mobiliteit) en verder

	Hooft	Laag	Hooft	Laag	Totaal	Hooft	Laag	Totaal
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	1	0:00:00	0,00
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	1	0:00:00	0,00
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	1	0:00:00	0,00
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	1	0:00:00	0,00
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	2	0:00:00	0,00
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	2	0:00:00	0,00
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	1	0:00:00	0,00
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	1	0:00:00	0,00
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	1	0:00:00	0,00
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	2	0:00:00	0,00
0,00%	0,00%	100,00%	100,00%	0,00%	0,00%	2	0:00:00	0,00

resultaten we  
Telpunt  
Delden, sluis (081)  
1 tot 1 van 1 resultaten (gefilter

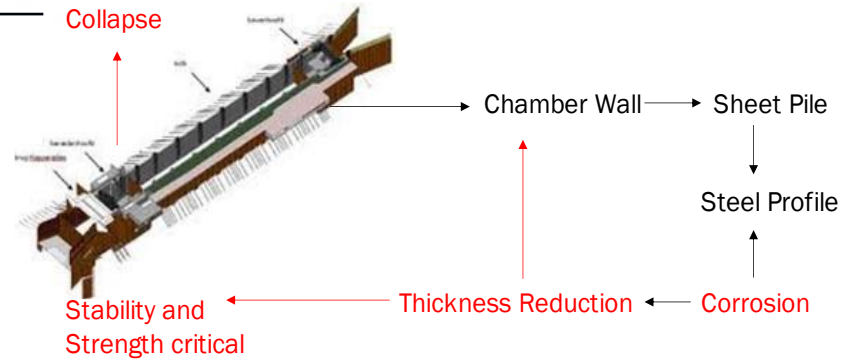
Gewicht	Som van Beschikbaarheid 6jg		Som van Beschikbaarheid 7jg		Som van Beschikbaarheid 8jg		Totaal Aantal in Norm		Totaal					
	Hooft	Laag	Hooft	Laag	Hooft	Laag	P Min	P Max	P Gem	P 85	P 90	P 95	P 99	
0							2089	103	1079	378,11	528	582	665	859
0							3407	136	1436	614,22	925	1034	1179	1378
0							787	112	1150	354,97	478	523	615	895
0							6199	138	1439	741,63	1110	1198	1307	1414
0							923	103	985	326,80	459	510	599	786
0							121	145	1023	418,42	602	679,5	775	1023
0							6161	120	1427	459,45	667	753	888	1175
0							287	121	1189	319,23	436,5	500,5	620,5	916

Sluis Eefde	540	3698	9518
Sluis Hagestein	450	135	5060
Sluis Hengelo	540	92	614
Sluis Weurt	504	974	20794
Spoldersluis	540	241	3813

# Rule 1: Technical ageing



Replace Walls



How many locks and which one have same condition data assigned?

## Rule 2: Technical ageing



Obsolete and Out  
of the stock ←



How many locks that have the gates registered as obsolete and out of the stock?

# Rule: Functional ageing rule



Widen the chamber  
Add new chamber



CEMT class not  
suitable



▲ De brug bij Macharen stond zaterdagochtend lang open voor de doorvaart van de Galactica. Hier was het aan de zijanten passen en meten. © Gabor Heeres/Foto Mallo

**Vijftien centimeter ruimte bij  
Hedelse spoorbrug is te weinig om  
de Galactica te laten passeren**

How many locks that became limited to provide these dimensions?

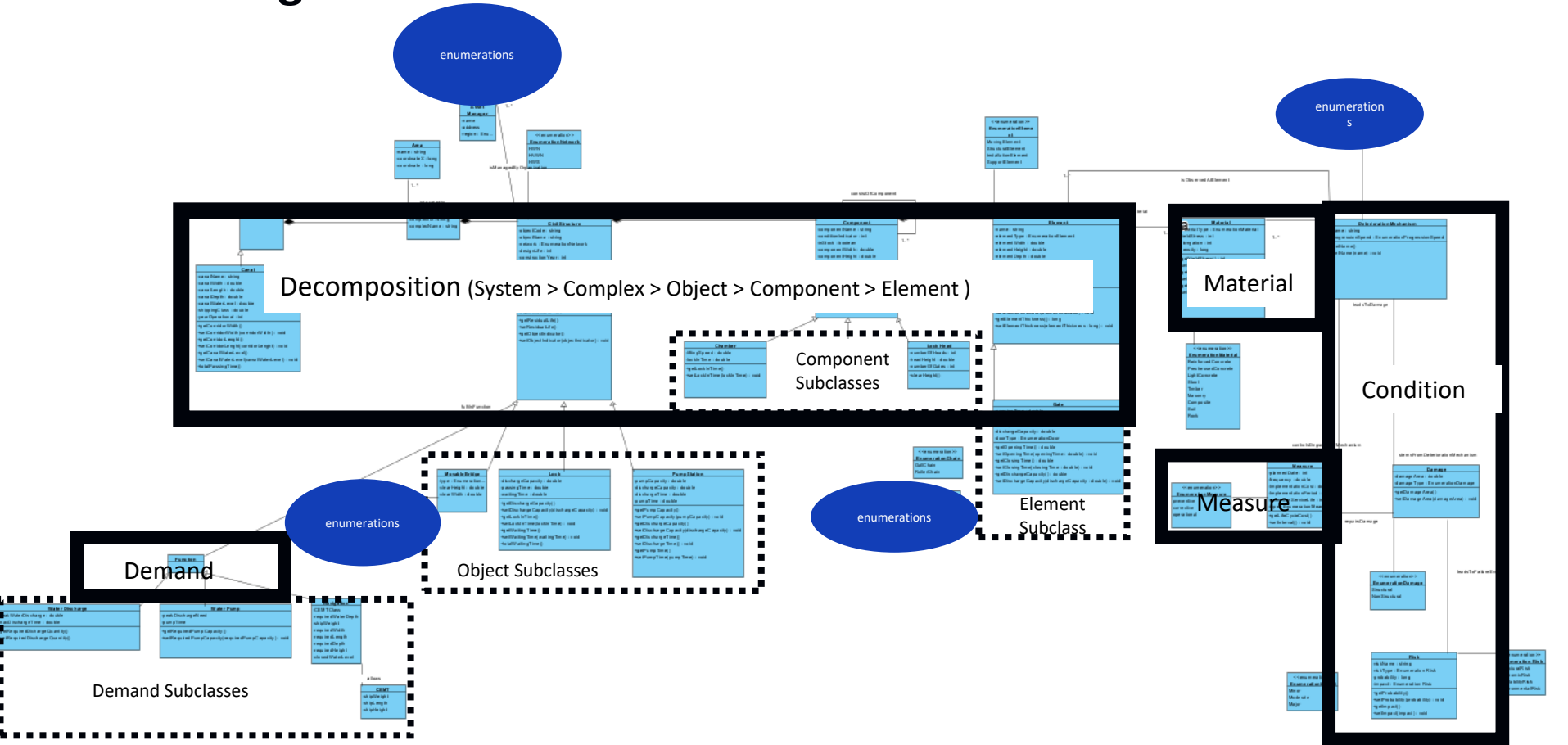
# Functional requirements (Shortened)

- How many navigation locks are operational the in the Netherlands?
- What are the functional demands on navigation locks on corridor and component level?
- What are the critical metrics on navigation locks for the demand and on what level can they be quantified?
- What are the critical components of navigation locks and what are their (sub)functions?
- What are the critical information for the navigation locks from asset managers' perspective?
- What are the condition information for navigation locks and what is a minimum set of information to be assigned to locks in terms to create an overview?

Examples on detailed CQs:

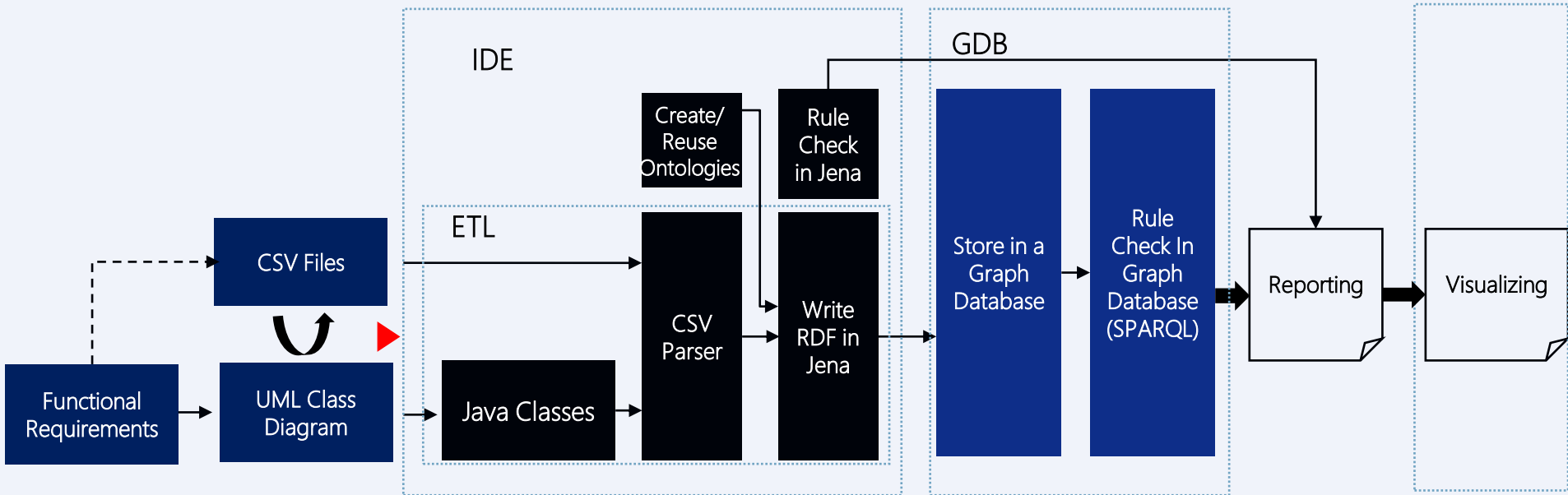
- Which navigation locks are monumental?
- How many navigation locks that have gates obsolete and not in-stock?
- What are the CEMT class types of navigation locks, what are the required dimension per CEMT class?
- What are the dimensions of the locks and also their critical components e.g. chambers, waiting area, lock-head?
- Is the CEMT class of the corridor the same with the CEMT class of the navigation locks?
- What is the waiting time for ships defined by the norm and measured at the locks?
- What are the other related objects of a complex, what are their critical properties e.g. clear height that may limit the ship dimensions that are essential function of lock?

# UML Diagram



# The route of using the Semantic Web Technologies

## Use of Apache Jena Framework



# Java Classes

- Open source tooling (Jena Framework) to create model.

The screenshot shows an IDE with a project explorer on the left containing 15 Java classes: Area.java, AssetManager.java, CEMClass.java, Chamber.java, CivilStructure.java, Complex.java, Component.java, Corridor.java, Damage.java, DeteriorationMechanism.java, Element.java, EnumerationChain.java, EnumerationDamage.java, EnumerationDoor.java, EnumerationElement.java, EnumerationFunction.java, EnumerationGate.java, EnumerationImpact.java, EnumerationMaterial.java, EnumerationMeasure.java, EnumerationMovabilityBridge.java, EnumerationNetwork.java, EnumerationProgressionSpeed.java, EnumerationRegion.java, EnumerationRisk.java, EnumerationRisk.java, Function.java, Gate.java, Installation.java, Lock.java, LockHead.java, Man.java, Material.java, MaterialType.java, Measure.java, MobilityBridge.java, Navigation.java, Property.java, PumpStation.java, Risk.java, RWISystem.java, WaitingArea.java, WaterDischarge.java, and WaterPump.java. The code editor on the right shows two methods: `toString()` and `add_component_as_resource()`. The `toString()` method returns a string with properties like `Object Code`, `Component Name`, `Component ID`, `Component's Main Material`, `Condition Indicator`, `Condition Indicator + Component Stock Level`, `instock`, `Component Width`, `Component Height`, `Component Depth`, `Component Length`, and `Damage Ids`. The `add_component_as_resource()` method initializes a `Model` with properties for `componentName`, `componentID`, `componentMainMaterial`, `conditionIndicator`, `instock`, `componentWidth`, `componentDepth`, `hasDamage`, and `Damage`. It also creates a `DamageSet` resource and adds `Damage` resources for each component.

Corridor, Complex, Object. Component. Element, Damage, Risk, Demand etc.

Data Properties

Object Properties

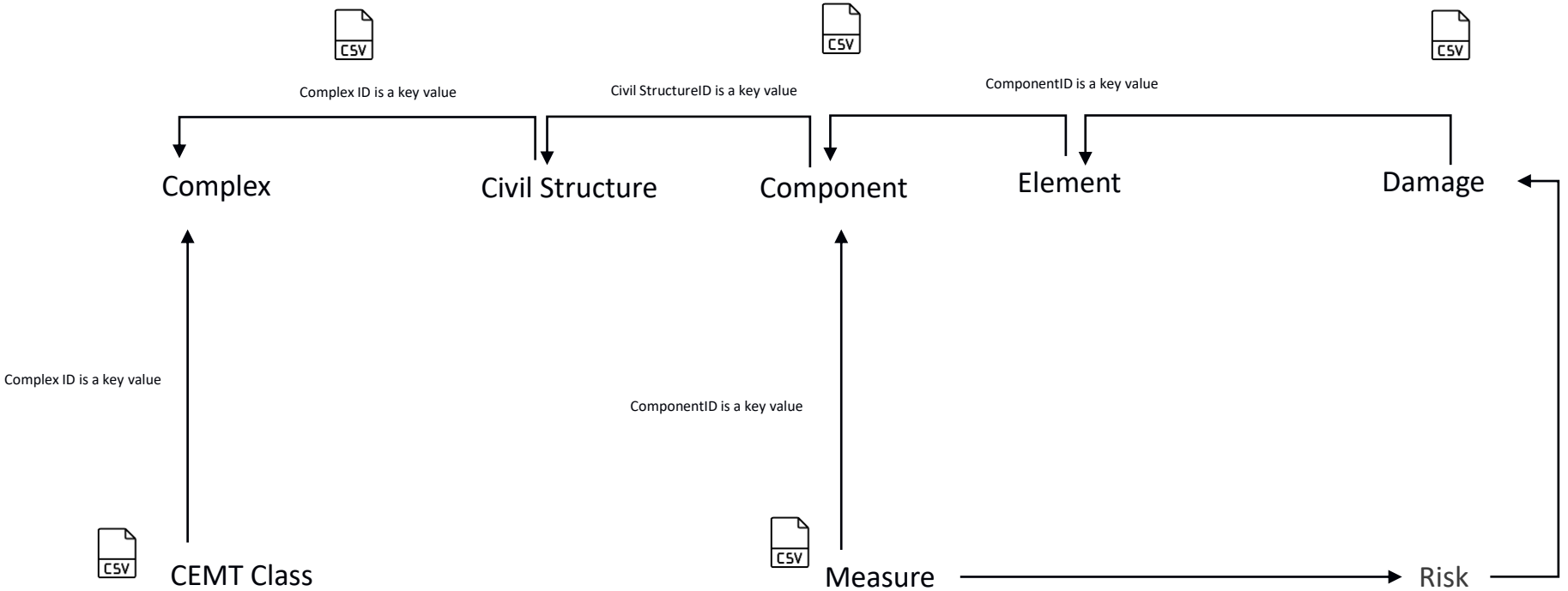
Methods

Initialization of Classes  
(Data read from 5 CSV files)

15 Java Classes  
& Parsing data  
(200 locks, 1000  
components...)



# Proof of concept: 3 rules, 5 Classes



# Using Jena to create triples

Damage added to Component

```

public String toString() {
    String damage_ids = get_damage_ids_as_string();
    {
        return "[Object Code = " + objectCode + ", Component Name=" + componentName + ", Component ID=" + componentID
            + ", Component's Main Material = " + componentMainMaterial + ", Condition Indicator="
            + conditionIndicator + ", Component Stock Level=" + inStock + ", Component Width" + componentWidth
            + ", Component Height" + componentHeight + ", Component Depth" + componentDepth
            + ", Component Length" + componentLength + ", Damage ids = " + damage_ids + " ]";
    }
}

public String get_damage_ids_as_string() {
    String damage_ids;
    if (Damages.size() > 0) {
        damage_ids = "[";
        for (Damage D : Damages) {
            damage_ids += D.damageID + " ";
        }
        damage_ids = damage_ids + " ]";
    } else {
        damage_ids = "[";
    }
    return damage_ids;
}

public Model add_component_as_resource(String ns, Model model) {
    String nsOnt = "http://example.org/KeBoTune";
    Property componentName = model.createProperty(nsOnt + "componentName");
    Property componentID = model.createProperty(nsOnt + "componentID");
    Property componentMainMaterial = model.createProperty(nsOnt + "componentMainMaterial");
    Property conditionIndicator = model.createProperty(nsOnt + "conditionIndicator");
    Property inStock = model.createProperty(nsOnt + "inStock");
    Property componentWidth = model.createProperty(nsOnt + "componentWidth");
    Property componentDepth = model.createProperty(nsOnt + "componentDepth");
    Property componentLength = model.createProperty(nsOnt + "componentLength");
    Property hasDamage = model.createProperty(nsOnt + "Damage");
    Damage damage;
    // Property DamageSet = model.createProperty("DamageSet");
    // Resource componentDamages = model.getResource(ns + componentName +
    // "damages");
    for (Damage D : Damages) {
        model.createResource(ns + "Damage" + D.damageID);
        model.add(model.getResource(ns + componentID), hasDamage, model.getResource(ns + "Damage" + D.damageID));
        // componentDamages.addLiteral(hasDamage, D.get_damage_as_resource(ns, model));
        // // I need to define this in Damage Class
        D.create_damage_with_data_properties(ns, model);
    }
    model.getResource(ns + componentID).addProperty(RDFS.Label, componentName)
        .addLiteral(componentID, componentID).addLiteral(componentMainMaterial, componentMainMaterial)
        .addLiteral(conditionIndicator, conditionIndicator).addLiteral(inStock, inStock)
        .addLiteral(componentWidth, componentWidth).addLiteral(componentDepth, componentDepth)
        .addLiteral(componentLength, componentLength).addProperty(hasDamage, hasDamage);
    return model;
}

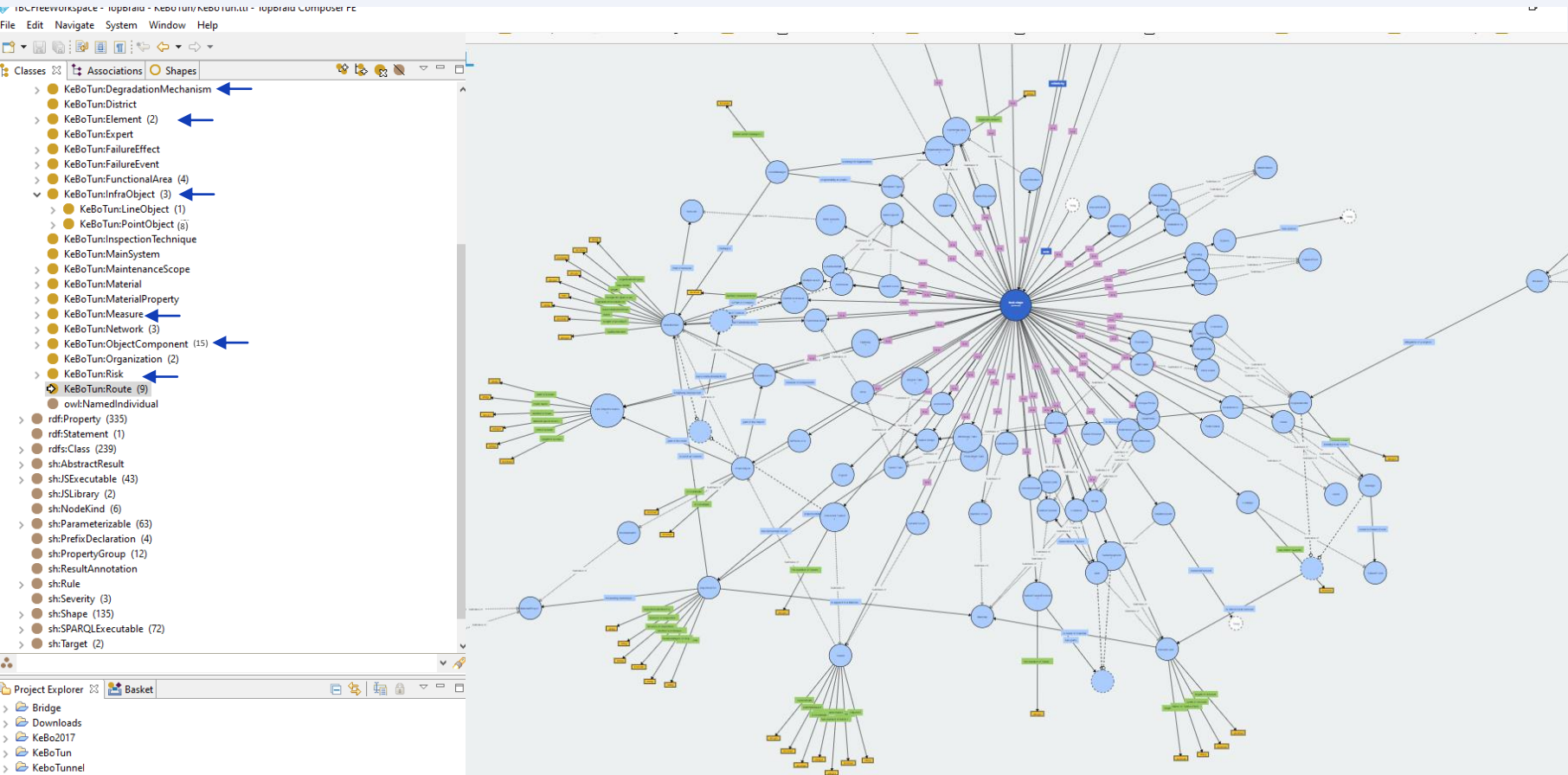
```

Selected Classes for PoC

- Area.java
- AssetManager.java
- CBMFCass.java
- Chamber.java
- CivilStructures.java
- Complex.java
- Component.java
- Conidor.java
- Damage.java
- DetonationMechanism.java
- Element.java
- EnumerationChain.java
- EnumerationDamage.java
- EnumerationOccur.java
- EnumerationElement.java
- EnumerationFunction.java
- EnumerationGate.java
- EnumerationImpact.java
- EnumerationMaterial.java
- EnumerationMeasure.java
- EnumerationMovability.java
- EnumerationNetwork.java
- EnumerationProgressionSpeed.java
- EnumerationRegio.java
- EnumerationRisk.java
- Function.java
- Gate.java
- Installation.java
- Lock.java
- LockHead.java
- Man.java
- Material.java
- MaterialType.java
- Measure.java
- MovabilityBridge.java
- Navigation.java
- Property.java
- PumpStation.java
- Risk.java
- RWSSystem.java
- WaitingLine.java
- WaterDischarge.java
- WaterPump.java

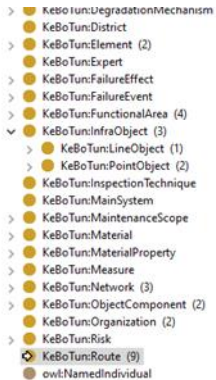
- Getting damage properties through key values (e.g. ID's)
- Add component as resource and create its properties via KeBoTune ontology
- Create damage as resources and add properties (by getting component resource)
- Get component resource and add properties

# Reusing KeBoTun



# Creating the RDF file (just where we are)

- First output as RDF – theoretically it should work
- Some things need to be fixed

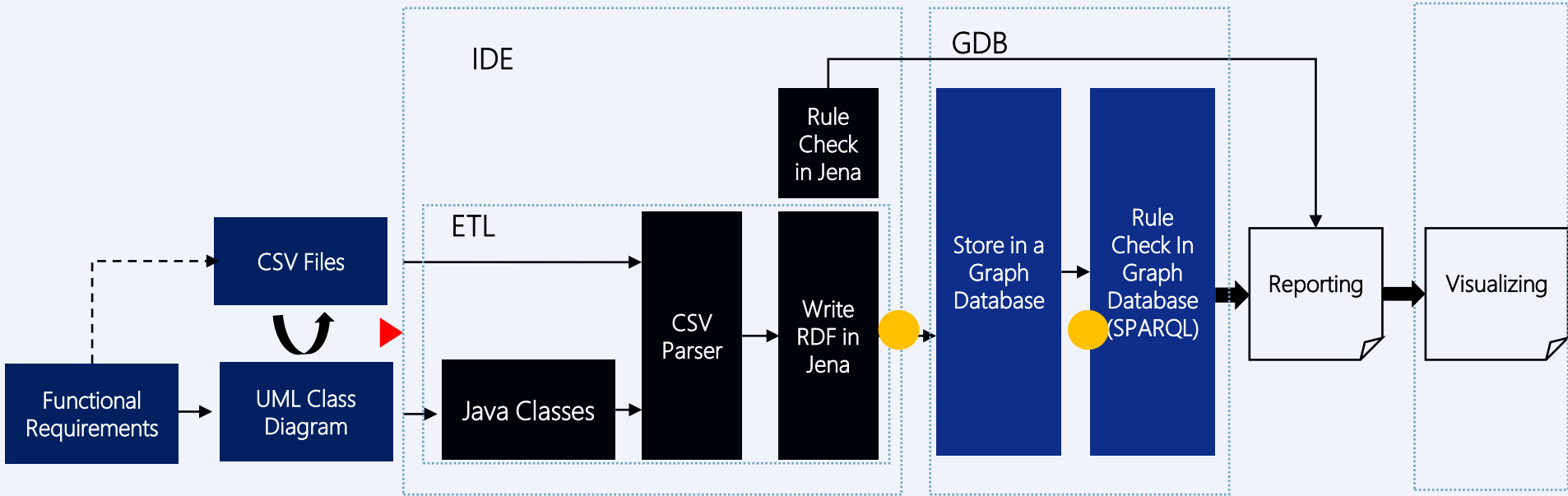


```
File Edit Format View Help
:https://www.keBo_sluices.nl#10B-001-05>
<http://www.w3.org/2000/01/rdf-schema#label>
  "Lorentz schutsluizen westkolk" ;
<https://www.kebo_sluices.nl/Ontology#constructionYear>
  "1931"^^<http://www.w3.org/2001/XMLSchema#long> ;
<https://www.kebo_sluices.nl/Ontology#damage>
  <https://www.keBo_sluices.nl#Damage113> ;
<https://www.kebo_sluices.nl/Ontology#length>
  "1555.0"^^<http://www.w3.org/2001/XMLSchema#double> ;
<https://www.kebo_sluices.nl/Ontology#manager>
  "MN District Noord" ;
<https://www.kebo_sluices.nl/Ontology#network>
  "HVMN" ;
<https://www.kebo_sluices.nl/Ontology#objectCode>
  "10B-001-05" ;
<https://www.kebo_sluices.nl/Ontology#objectHistory>
  "Rood" ;
<https://www.kebo_sluices.nl/Ontology#qualityIndicator>
  "5"^^<http://www.w3.org/2001/XMLSchema#long> ;
<https://www.kebo_sluices.nl/Ontology#type>
  "Schutsluizen" ;
<https://www.kebo_sluices.nl/Ontology#width>
  "0.0"^^<http://www.w3.org/2001/XMLSchema#double> .

:https://www.keBo_sluices.nl#Damage114>
<hasDamageArea>  "2.0"^^<http://www.w3.org/2001/XMLSchema#double> ;
<hasDamageType>  "Poor / insufficient functioning" .

:https://www.kebo_sluices.nl#45B-001-01>
<http://www.w3.org/2000/01/rdf-schema#label>
  "Sluis St. Andries" ;
<https://www.kebo_sluices.nl/Ontology#constructionYear>
  "1934"^^<http://www.w3.org/2001/XMLSchema#long> ;
<https://www.kebo_sluices.nl/Ontology#damage>
  <https://www.keBo_sluices.nl#Damage114> .
```

# Where we are, where to move forward



# Take aways until now, Moving Forward

- **This is a knowledge development project to explore the SW with no dependency for 1 specific decision moment.**
  - Accessing the data from its source is ideal, this is to prove a concept.
- **The major issue was that there were limited actual examples (except Jena documentation).**
  - Once the classes (+parsed data) were linked, Jena part was very intuitive (even though the project is not finished).
- **Most essential decision information comes from programming inspection reports/datasets. Yet that are almost always overlooked.**
- **There is always a top model, instantiated to many object types as *point-object* civil structures.**
  - The 5 main classes (and reused KeBoTun Ontology) have proven that we may need much simpler model for condition definition of the civil infrastructure.
- **There is always typical failure mechanisms associated to object types.**
- **Any additional properties can be defined via e.g. DOT Ontology.**
- **The PoC will be extendible for any civil structure as the data structures we use don't change and top-model don't change.**
  - But next step is to finish the PoC, and move to more ideal situation step by step.
- **Meanwhile we tag along the real assessor/decision maker, who became very enthusiastic ("*Inspection data-template*")**
- **There is a lot to gain and have real impact on the society, which is shadowed by other societal trends.**

A nighttime photograph of a city street featuring a modern, curved pedestrian bridge with a glass railing. The bridge is illuminated with warm lights, and its reflection is visible in the water below. In the background, there are several multi-story buildings with lit windows, and a prominent curved building with a glass facade. The scene is captured with a long exposure, resulting in vibrant green and yellow light trails from moving vehicles or lights, creating a sense of motion and energy. The overall atmosphere is urban and contemporary.

› **THANK YOU.**

For more questions:

[esra.bektas@tno.nl](mailto:esra.bektas@tno.nl)

**TNO** innovation  
for life