

# Knowledge-graph-based Digital Twin for network

Journées du GDR RSD – 19<sup>th</sup> and 20<sup>th</sup> March

Pauline Folz - PhD – Orange Research

Fano Ramparany – PhD – Orange Research

Thomas Hassan – PhD – Orange Research

Maria Massri – PhD – Orange Research

Fabrice Blache – Orange Research

Thierry Coupaye – Head of Augmented Operations Research Domain

Sébastien Bolle – Twin Modeling & Semantics Research Program Manager



# Autonomous network levels

**Autonomous networks levels**

Level Definition	L0: Manual Operation & Maintenance	L1: Assisted Operation & Maintenance	L2: Partial Autonomous Network	L3: Conditional Autonomous Network	L4: High Autonomous Network	L5: Full Autonomous Network
Execution	P	P/S	S	S	S	S
Awareness	P	P	P/S	S	S	S
Analysis	P	P	P	P/S	S	S
Decision	P	P	P	P/S	S	S
Intent/Experience	P	P	P	P	P/S	S
Applicability	N/A	Select scenarios				All scenarios

P: Personnel, S: Systems

Figure 1: Autoums networks levels form TM Forum [1]

[1] TM Forum [Whitepaper](#): Autonomous Networks: Empowering Digital Transformation for the Telecoms Industry

# Context

Multiple data sources to operate and manage the network

Multiple vendor equipment



# Problem statement

How can a **unified network topology** be represented and searched?

# Digital Twin

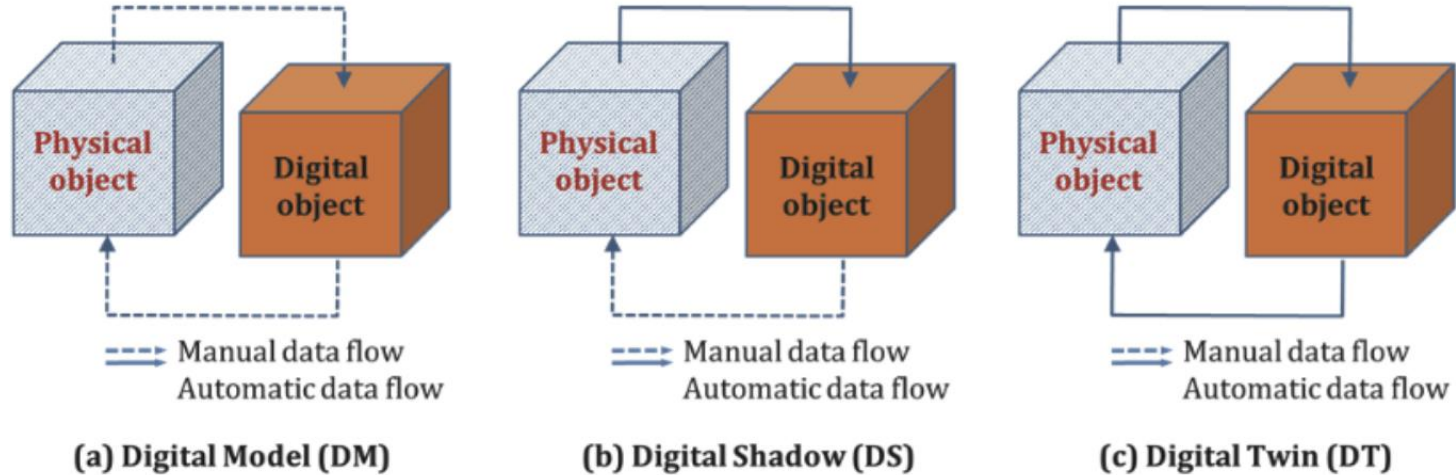


Figure 2: Taxonomy of digital representation [2] [3]

[2] GRIEVES, Michael. Digital model, digital shadow, digital twin. Preprint, 2023.

[3] FULLER, Aidan, FAN, Zhong, DAY, Charles, et al. Digital twin: Enabling technologies, challenges and open research. IEEE access, 2020, vol. 8.

# Knowledge Graph

- Use of ontologies to model the data
  - Ontology: formal and shared representation of data schema
  - Data can be modelled with several ontologies
- Complex query: path(s) between two network nodes given some properties
- Filtering queries & query rewriting

# Proposal

Use a knowledge-graph-based digital twin to model the network topology

# Network data: Yang

**Network protocol:** Netconf/restconf uses Yang modes for configuration and management.

YANG modules define the **structure** of the network data and enable **standardized interactions**.

Example of a ietf's Yang module:

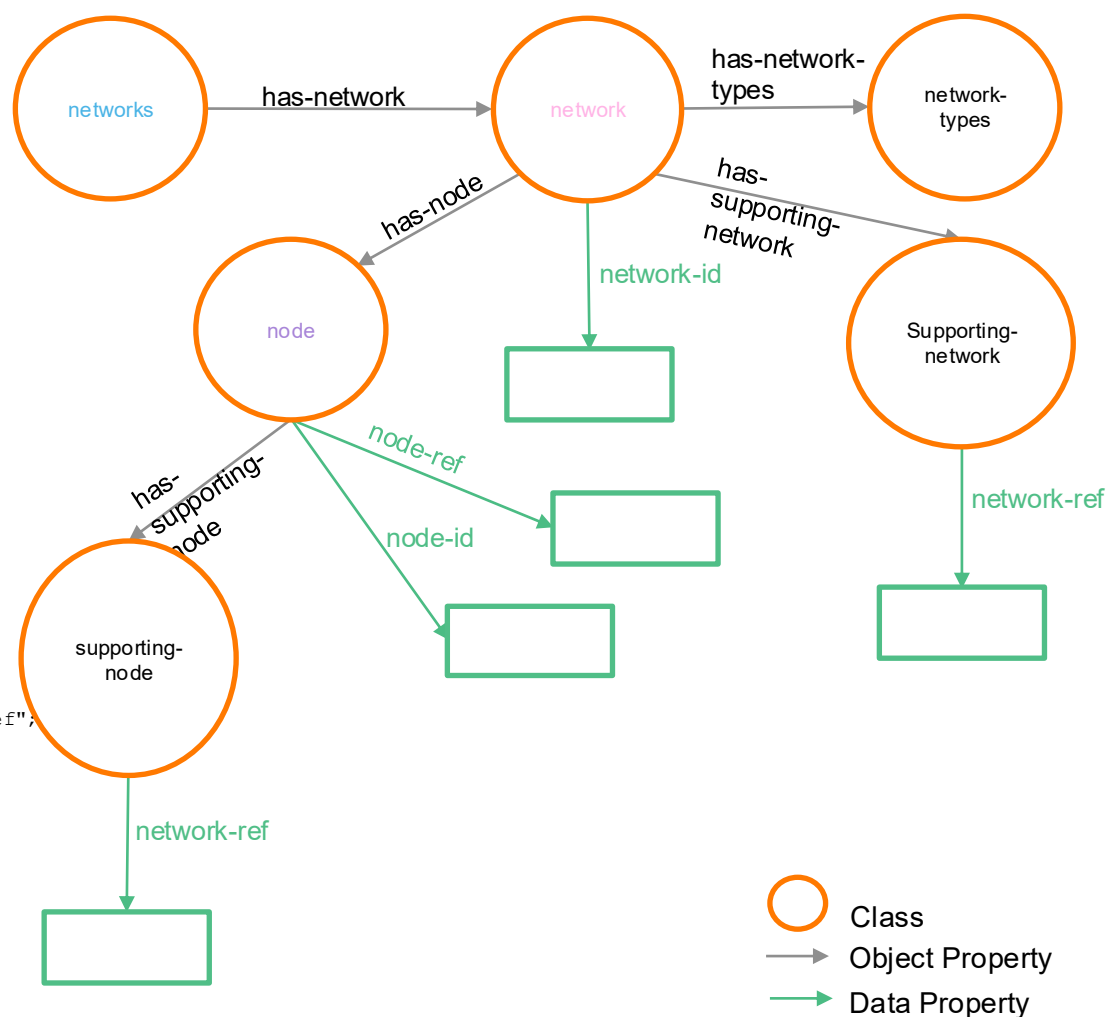
```
container networks {
  list network {
    key "network-id";
    leaf network-id {
      type network-id;
    }
  }
  container network-types {}
  list supporting-network {
    key "network-ref";
    leaf network-ref {
      type leafref {
        path "/nw:networks/nw:network/nw:network-id";
        require-instance false;
      }
    }
  }
}
list node {
  key "node-id";
  leaf node-id {
    type node-id;
  }
  list supporting-node {
    key "network-ref node-ref";
    leaf network-ref {
      type leafref {
        path "../..../nw:supporting-network/nw:network-ref";
        require-instance false;
      }
    }
  }
  leaf node-ref {
    type leafref {
      path "/nw:networks/nw:network/nw:node/nw:node-id";
      require-instance false;
    }
  }
}
}
```

# Example – ietf-network

```

container networks {
  list network {
    key "network-id";
    leaf network-id {
      type network-id;
    }
  }
  container network-types {}
  list supporting-network {
    key "network-ref";
    leaf network-ref {
      type leafref {
        path "/nw:networks/nw:network/nw:network-id";
        require-instance false;
      }
    }
  }
}
list node {
  key "node-id";
  leaf node-id {
    type node-id;
  }
  list supporting-node {
    key "network-ref node-ref";
    leaf network-ref {
      type leafref {
        path "../..../nw:supporting-network/nw:network-ref";
        require-instance false;
      }
    }
  }
  leaf node-ref {
    type leafref {
      path "/nw:networks/nw:network/nw:node/nw:node-id";
      require-instance false;
    }
  }
}
}
}

```



# Process flow

1. Sample network with 5 nodes
2. Build data schema
3. Integrate data: network topology
4. Build Digital Twin
5. Visualize Network topology

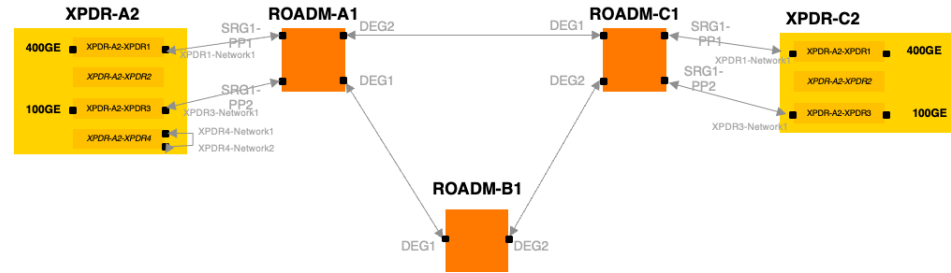
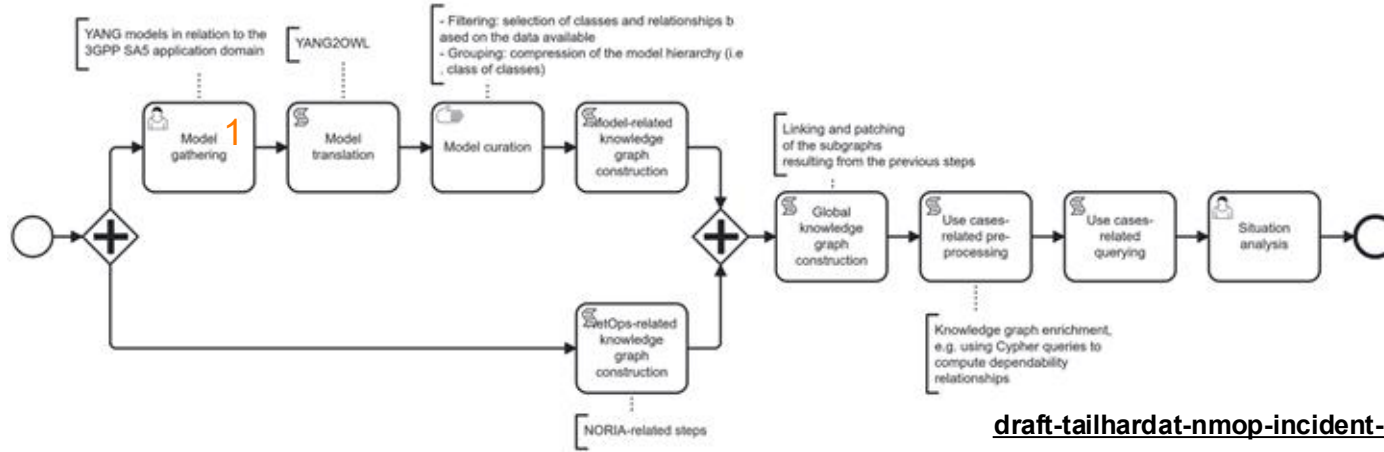


Figure 3: 5 nodes network sample

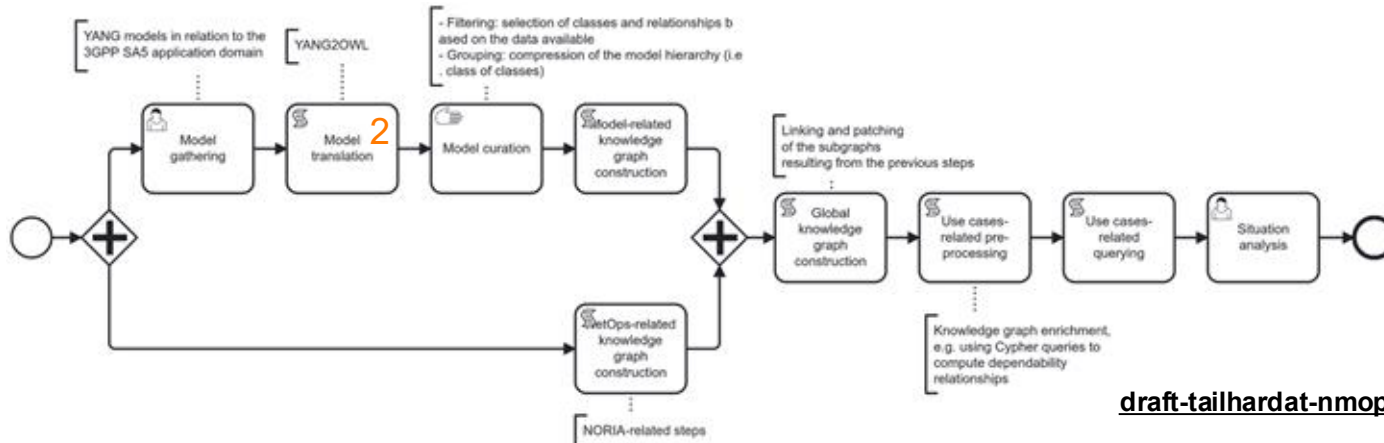


# Process flow

1. Sample network with 5 nodes
2. **Build data schema**
3. Integrate data: network topology
4. Build Digital Twin
5. Visualize Network topology

## Yang-2-Owl:

- Gather Yang model
- Build ontologies



# Process flow

1. Sample network with 5 nodes
2. Build data schema
3. Integrate data: network topology
4. Build Digital Twin
5. Visualize Network topology

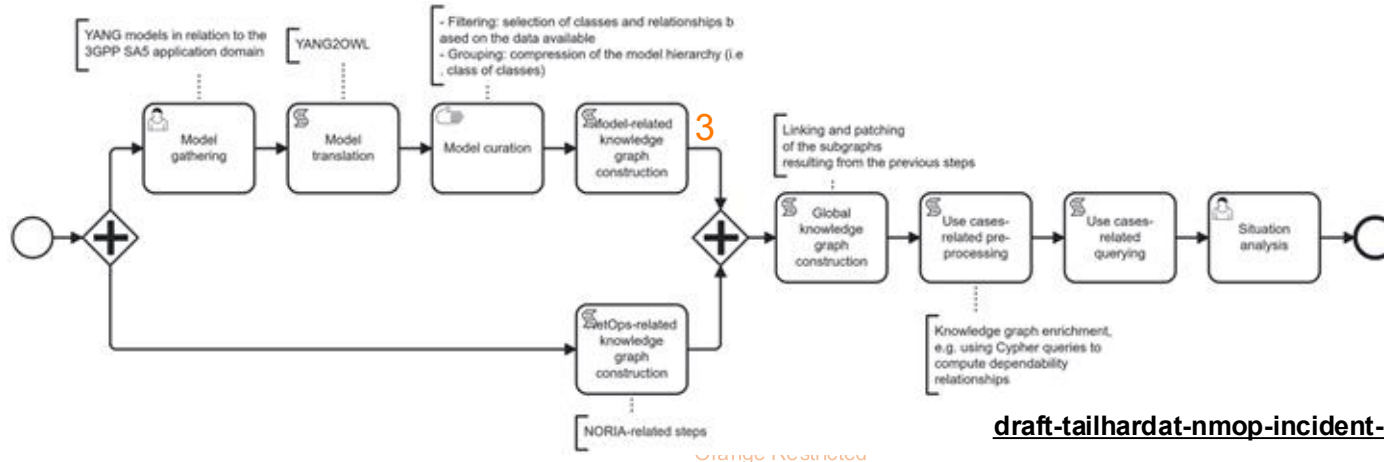
## Json-2-RDF:

### Input:

- Ontology (from previous step)
- Yang conf.

### Conversion:

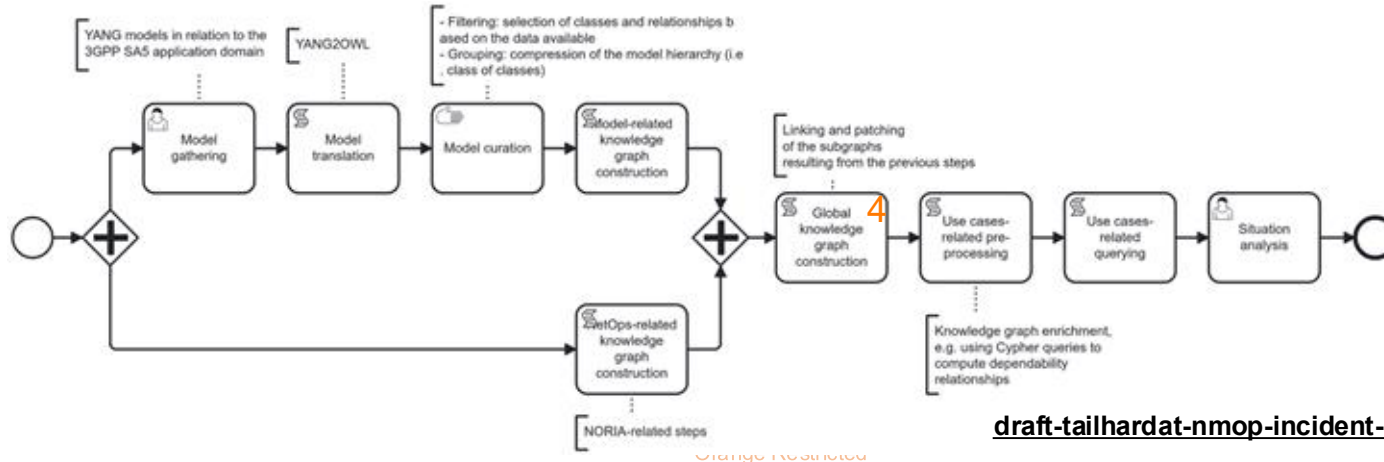
- Use identifier to match class, object property or data property
- Create relation for leafref



# Process flow

1. Sample network with 5 nodes
2. Build data schema
3. Integrate data: network topology
4. Build Digital Twin
5. Visualize Network topology

Thing in the future: graph-based Digital Twin platform from Orange Research



# Process flow

1. Sample network with 5 nodes
2. Build data schema
3. Integrate data: network topology
4. Build Digital Twin
5. Visualize Network topology

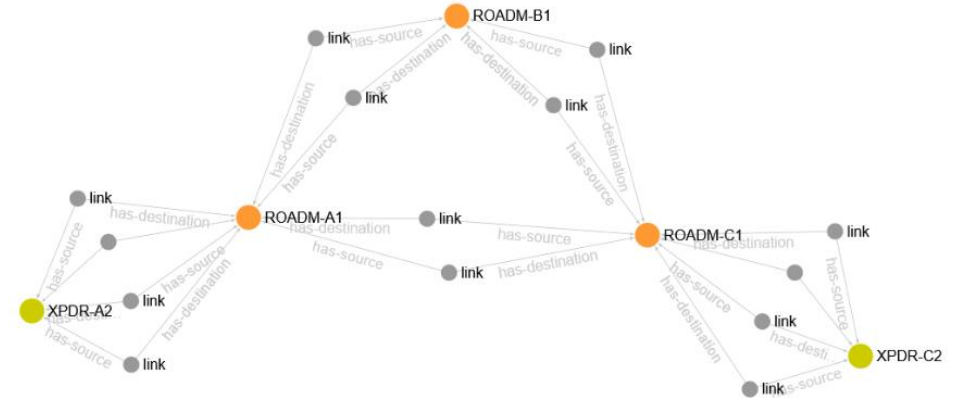
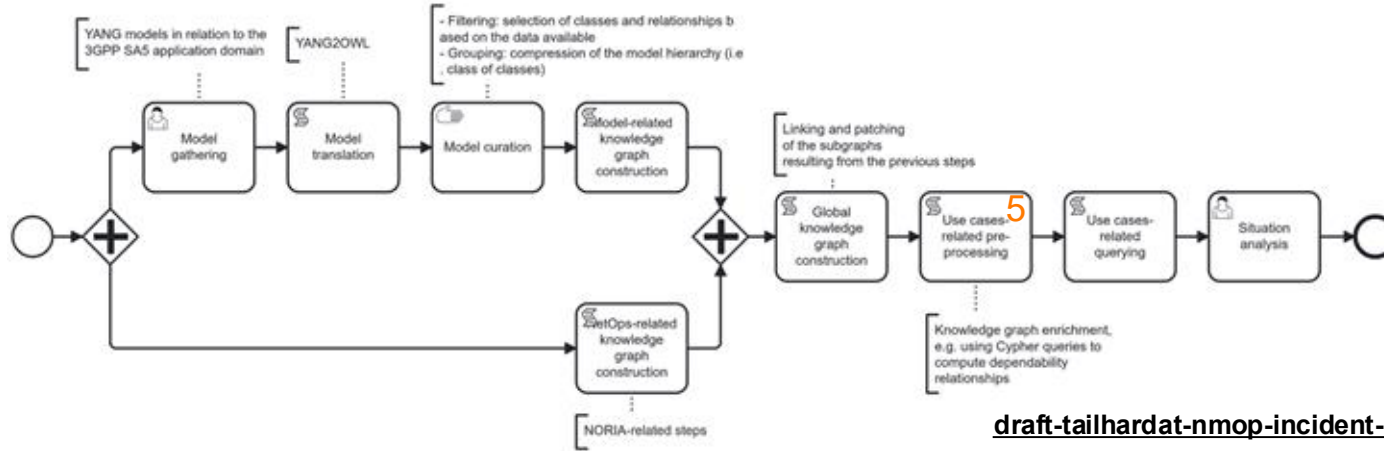


Figure 4: Digital Twin of 5 nodes network



# Conclusion & Perspective

## Yang2Owl:

- Allow to build a unified schema for the network

## Json2Rdf:

- Build a semantic digital twin of the network topology

## Perspectives:

- Discuss Yang2Owl & Json2Rdf into working group (ietf, etsi)
- Demonstration on an next generation optical transport network (i.e, all-photonic network from IOWN Global Forum)
- Integrate cognitive functions to the semantic digital twin (simulation and prediction)

[pauline.folz@orange.com](mailto:pauline.folz@orange.com)  
[sebastien.bolle@orange.com](mailto:sebastien.bolle@orange.com)

# Thank you