Network Management Operations L. Tailhardat
Internet-Draft Orange Research
Intended status: Informational R. Troncy
Expires: 16 November 2025 EURECOM
Y. Chabot
F. Ramparany
P. Folz
Orange Research
15 May 2025

Knowledge Graphs for Enhanced Cross-Operator Incident Management and Network Design
draft-tailhardat-nmop-incident-management-noria-02

Abstract

Operational efficiency in incident management on telecom and computer networks requires correlating and interpreting large volumes of heterogeneous technical information. Knowledge graphs can provide a unified view of complex systems through shared vocabularies. YANG data models enable describing network configurations and automating their deployment. However, both approaches face challenges in vocabulary alignment and adoption, hindering knowledge capitalization and sharing on network designs and best practices. To address this, the concept of a IT Service Management (ITSM) Knowledge Graph (KG) is introduced to leverage existing network infrastructure descriptions in YANG format and enable abstract reasoning on network behaviors. The key principle to achieve the construction of such ITSM-KG is to transform YANG representations of network infrastructures into an equivalent knowledge graph representation, and then embed it into a more extensive data model for Anomaly Detection (AD) and Risk Management applications. In addition to use case analysis and design pattern analysis, an experiment is proposed to assess the potential of the ITSM-KG in improving network quality and designs.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://genears.github.io/draft-tailhardat-nmop-incident-management-noria/draft-tailhardat-nmop-incident-management-noria.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-tailhardat-nmop-incident-management-noria/.

Discussion of this document takes place on the Network Management Operations Working Group mailing list (mailto:nmop@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/nmop/. Subscribe at https://www.ietf.org/mailman/listinfo/nmop/.

Source for this draft and an issue tracker can be found at https://github.com/genears/draft-tailhardat-nmop-incident-management-noria.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Incident management on telecom and computer networks, whether it is related to infrastructure or cybersecurity issues, requires the ability to simultaneously and quickly correlate and interpret a large number of heterogeneous technical information sources. Knowledge graphs, by structuring heterogeneous data through shared vocabularies, enable providing a unified view of complex technical systems, their ecosystem, and the activities and operations related to them (see [I-D.marcas-nmop-knowledge-graph-yang] and [NORIA-O-2024]). Using such formal knowledge representation allows for a simplified interpretation of networks and their behavior, both for NetOps & SecOps teams and artificial intelligence (AI) algorithms (e.g. anomaly detection, root cause analysis, diagnostic aid, situation summarization), and paves the way, in line with the Network Digital Twin vision [I-D.irtf-nmrg-network-digital-twin-arch], for the development of tools for detecting and analyzing complex network incident situations through explainable, actionable, and shareable models (see [FOLIO-2018], [SLKG-2023], and [GPL-2024]).

However, despite potential benefits of using knowledge graphs, these are not mainstream yet in commercial network deployment systems and decision support systems (see [NORIA-UI-2024] for more on the decision support systems perspective). YANG is a widely used standard among operators for describing network configurations and automating their deployment. Using YANG representations in the form of a KG, as suggested in [I-D.marcas-nmop-knowledge-graph-yang], would minimize the effort required to adapt network management tools towards the unified vision and applications evoked above. The lack of alignment between various YANG models on key concepts (e.g. for describing network topology) is, however, hindering this evolution [I-D.boucadair-nmop-rfc3535-20years-later].

Furthermore, although [I-D.netana-nmop-network-anomaly-lifecycle] addresses the capitalization of incident management knowledge through a YANG model, it can be observed that the overall scope of YANG models does not naturally cover the description of the networks' ecosystem (e.g. physical equipment location, operator organization, supervision systems) or the description of network operations from an IT service management (ITSM) perspective (e.g. business processes and design rules used by the company, scheduled modification operations, remediation actions performed during incident handling). As a consequence, the continuous improvement of network quality & designs requires additional data cross-referencing operations to properly contextualize incidents and learn from remediation actions taken (e.g. analyzing intervention technicians' verbatim, comparing actions performed on similar incidents but occurring on different networks). As a result of these additional efforts of contextualization, the capitalization of knowledge typically remains confined at the level of each network operator. This, in turn, hinders the sharing of information within the community of researchers and system designers regarding failure modes and best practices to adopt, considering the concept of overall improvement of IT systems and the Internet.

Realizing an ITSM knowledge graph for network deployment, anomaly detection and risk management applications has been studied for several years in the Semantic Web community (i.e. knowledge representation and automated reasoning leveraging Web technologies such as [RDF], [RDFS], [OWL], and [SKOS]). Among other examples: the DevOpsInfra ontology [DevOpsInfra-2021] allows for describing sets of computing resources and how they are allocated for hosting services; the NORIA-O ontology [NORIA-O-2024] allows for describing a network infrastructure & ecosystem, its events, diagnosis and repair actions performed during incident management. Assuming the continuous integration into a knowledge graph of data from ticketing systems, network monitoring solutions, and network configuration management databases, we remark that the resulting knowledge graph (Figure 1) implicitely holds the necessary information to (automatically) learn incident contexts (i.e. the network topology, its set of states and set of events prior to the incident) and remediation procedures (i.e. the set of actions and network configuration changes carried-out to resolve the incident).

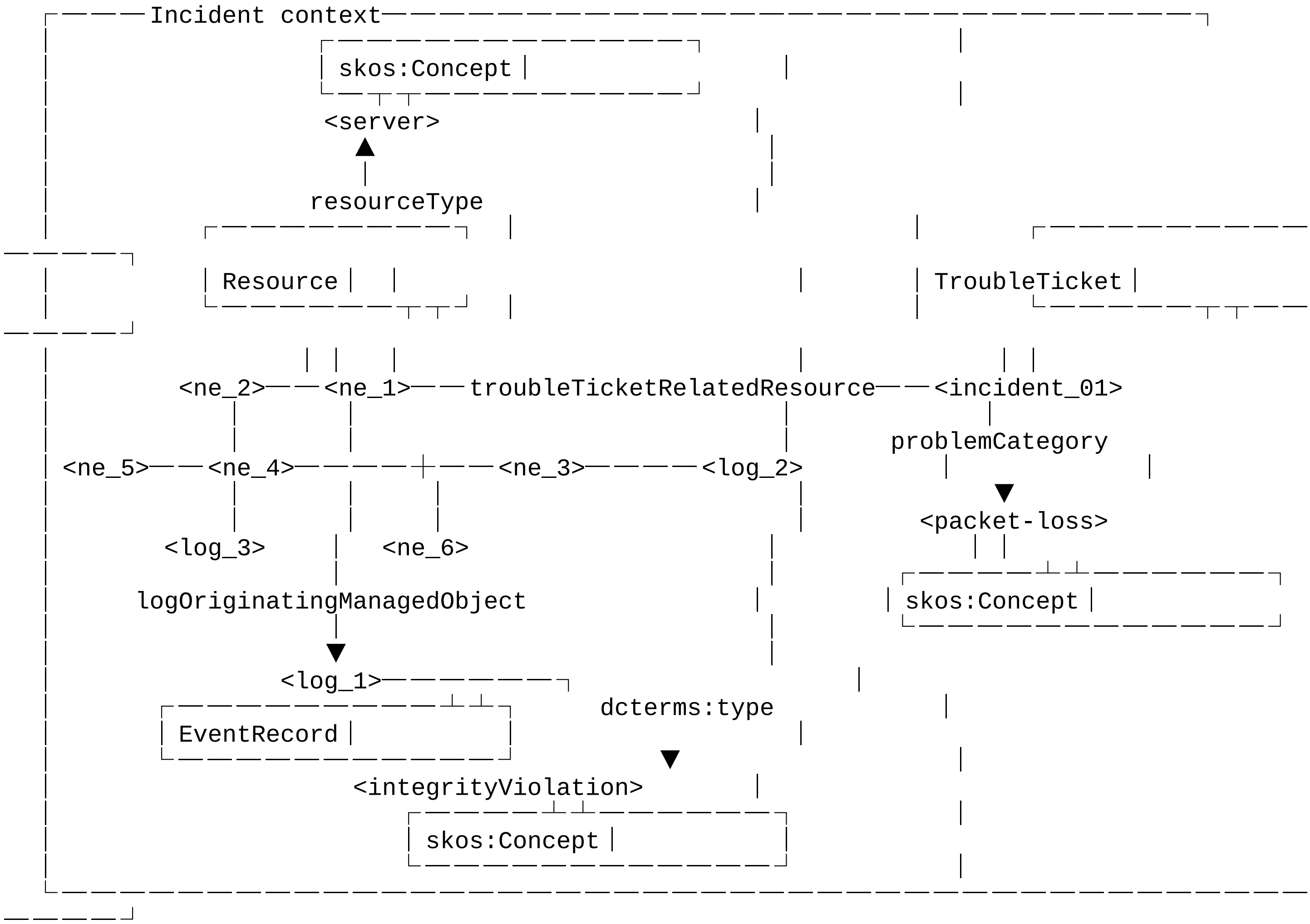

Figure 1: Learning an incident signature seen as a classification model that is trained on the relationship of the incident context (i.e. a subgraph centered around a Resource entity concerned by a given TroubleTicket) to the problem class defined at the TroubleTicket entity level. Arrows are for object properties (owl:ObjectProperty), double line edges are for object class relationships (rdf:type).

By going a step further, we notice that a generic understanding of incident context can be extracted and shared among operators from knowledge graphs. Indeed, a knowledge graph, being an instantiation of shared vocabularies (e.g. RDFS/OWL ontologies and controlled vocabularies in SKOS syntax), sharing incident signatures can be done without revealing infrastructure details (e.g. hostname, IP address), but rather the abstract representation of the network (i.e. the class of the knowledge graph entities and relationships, such as "server" or "router", and or "IPoWDM link").

The remainder of this document is organized as follows. Firstly, the concept of an ITSM-KG is introduced in Section 3 towards leveraging existing network infrastructure descriptions in YANG format and enabling abstract reasoning on network behaviors. The relation of the ITSM-KG proposal to the Digital Map [I-D.havel-nmop-digital-map-concept] is notably discussed in this section. Secondly, strategies for the ITSM-KG construction are discussed in Section 4. This include YANG models transformation in Section 4.1, implementing alignments of models with the ITSM-KG in Section 4.2, and knowledge graph construction pipeline designs in Section 4.3. The Section 4.3 notably focuses on addressing the handling of event data streams and providing a unified view for different stakeholders, also known as the data federation architecture. Finally, an experiment is proposed in Section 5 to assess the potential of the ITSM-KG in improving network quality and designs. The implementation status related to this document is also reported in this section.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. An ITSM-KG for Learning and Sharing Network Behavioral Models

3.1. Principles

As evoked in Section 1, a detailed characterization of network behavior requires combining several facets of data related both to the configuration of the networks and to their lifecycle, as well as the ecosystem in which they are operated. In this document, we will consider the following fundamental definitions as a means to achieve the combination of all these facets of data in a convenient way, regardless of their origin, for operational efficiency in incident management and change management with the aid of AI tools:

ITSM-KG: A knowledge graph in RDFS/OWL syntax tha enables change management activities, anomaly detection, and risk analysis at the organizational level by combining heterogeneous data sources from the configuration data of the network’s structural elements, events occurring on this network, and any other data useful to the business for the effective management of the services provided by this network.

ONTO-ITSM: For a given ITSM-KG, the RDFS/OWL ontology that structures the ITSM-KG.

ONTO-YANG-MODEL: For a given YANG model, its equivalent RDFS/OWL representation.

ONTO-META: An ontology that contributes to structuring some ITSM-KG, regardless of the specifics of a given application domain or ITSM-KG instance, in the sense that it provides an abstract IT Service Management model (i.e. it holds generic concept and property definitions for realizing IT Service Management activities).

ONTO-LINKER: For a given (set of) ONTO-YANG-MODEL and a given ONTO-META, the implementation of the equivalence relationships between the key concepts and key properties of the (set of) ONTO-YANG-MODEL and ONTO-META.

Based on these definitions, which will be discussed in more detail later in this document, Figure 1 can be seen as an illustration of ITSM-KG from which a subgraph has been extracted, allowing for incident situation to be analyzed through querying. For example, close to ideas from [I-D.netana-nmop-network-anomaly-lifecycle], querying the evolution of network entities states from the ITSM-KG during some incident remediation stage could bring to identify the causal graph underlying incident resolution. As the querying would go through the ONTO-ITSM, the causal graph would de-facto be an abstraction of the situation, thereby enabling knowledge capitalization and sharing for similar incidents that could occur later.

3.2. Relation to the Digital Map

Similar to the concept of ITSM-KG discussed in this document, the concept of Digital Map discussed in [I-D.havel-nmop-digital-map-concept] emphasizes the need to structure heterogeneous data describing networks in order to simplify network management operations through unified access to this data. The ITSM-KG can be seen as a meta-knowledge graph that extends the Digital Map concept by adding information about the lifecycle of infrastructures and services, as well as the context of their usage. These additional pieces of information are considered essential for learning shareable activity models of systems.

To clarify this positioning, the following lists (Section 3.2.1, Section 3.2.2, and Section 3.2.3) reflect the compliance of the meta-KG concept with the Digital Map Requirements defined in [I-D.havel-nmop-digital-map-concept]. A symbol to the right of each requirement name indicates the nature of compliance: *+* for compatibility, */* for partial satisfaction, *-* for non-compliance with the requirement. A comment is provided as necessary.

3.2.1. Core Requirements

+ REQ-BASIC-MODEL-SUPPORT: nothing to report (n.t.r.)

+ REQ-LAYERED-MODEL: n.t.r.

/ REQ-PROG-OPEN-MODEL: Partially satifying the requirement as the concept of meta-KG mainly relate to the knowledge representation topic rather than to the platform running the Digital Map service on top of the meta-knowledge graph.

/ REQ-STD-API-BASED: Same remark as for REQ-PROG-OPEN-MODEL.

+ REQ-COMMON-APP: n.t.r.

+ REQ-SEMANTIC: n.t.r.

+ REQ-LAYER-NAVIGATE: n.t.r.

+ REQ-EXTENSIBLE: Knowledge graphs implicitly satisfy this requirement, notably with OWL [OWL] and SKOS [SKOS] constructs if considering RDF knowledge graphs for the meta-KG (e.g. owl:sameAs to relate a meta-KG entity to some other entity of another knowledge graph, owl:equivalentClass to link concepts and properties used to interpret the meta-KG to concepts and properties from other data models, skos:inScheme to group new items of a controled-vocabulary as part of a skos:ConceptScheme).

+ REQ-PLUGG: Same remark as for REQ-EXTENSIBLE.

+ REQ-GRAPH-TRAVERSAL: This capability is naturally enabled as the meta-KG concept involves using a graph data structure.

3.2.2. Design Requirements

- REQ-TOPO-ONLY: Requirement not satisfied as the meta-KG involves

to have more than topological data to interpret and contextualize the network behavior.

- REQ-PROPERTIES: Same remark as for REQ-TOPO-ONLY.

- REQ-RELATIONSHIPS: Same remark as for REQ-TOPO-ONLY.

+ REQ-CONDITIONAL: Native, notably considering the expressiveness of SPARQL [SPARQL11-QL] if using the Semantic Web protocol stack to run the meta-KG concept.

+ REQ-TEMPO-HISTO: n.t.r.

3.2.3. Architectural Requirements

+ REQ-DM-SCALES: This capability applies as we can use data aggregation at the graph level (Figure 10 and Figure 11 compared to Figure 8 and Figure 9), aggregation without loss of information (Figure 10 and Figure 11), and load balancing (horizontal scaling) by partitioning the meta-KG (Figure 12). Further, ease of integration is enabled thanks to existing standard graph data access protocols (e.g. SPARQL Federated Queries [SPARQL11-FQ], as illustrated in Figure 12).

/ REQ-DM-DISCOVERY: Same remark as for REQ-PROG-OPEN-MODEL.

4. Strategies for the ITSM-KG Construction

In this section, we firstly define in Section 4.1 two YANG-based data transformation scenario, namely the YANG-KG-SEMANTIC-EQUIVALENCE and YANG-KG-SEMANTIC-GENERALIZATION scenarios. The YANG-KG-SEMANTIC-GENERALIZATION scenario is then used as a basis in Section 4.2 to illustrate strategies to reuse YANG models transformed in RDFS/OWL syntax in a higher-level ontology that would structure the ITSM-KG. Finally, two Extract-Transform-Load (ETL) pipeline approaches and a data federation architecture are presented in Section 4.3 to meet the needs of constructing and exploiting the ITSM-KG.

4.1. From YANG-based Configurations to Meta-Knowledge Graph

In the following, we consider the use of Semantic Web technologies as the foundation for representing data in the form of a knowledge graph. We also assume the ability to transform a description of configurations and network infrastructures expressed accordingly to a given (set of) YANG model(s) into a knowledge graph representation.

For the realization of this data transformation, we identify the following scenarios:

YANG-KG-SEMANTIC-EQUIVALENCE: The ontology structuring the target knowledge graph is an exact equivalence of the many YANG models organizing the configuration data.

YANG-KG-SEMANTIC-GENERALIZATION: The ontology structuring the target KG is a generalization of the YANG models organizing the configuration data.

We note that the YANG-KG-SEMANTIC-EQUIVALENCE case requires a significant knowledge engineering effort to align all YANG models into a coherent ontology with a sufficient level of abstraction to enable the discovery and analysis of emergent behavioral models of networks independently of local configuration specifics. However, this case has the advantage of being relatively easy to implement based on the available configuration data of an operator, for example, by implementing [RML] rules for constructing a knowledge graph from this data.

For the YANG-KG-SEMANTIC-GENERALIZATION case, we observe that the transformation effort involves:

1. Being able to transform YANG models into their RDFS/OWL equivalent to provide a consistent interpretation of configuration data in a knowledge graph that aligns with each data source.

2. Being able to provide a generalized interpretation of these transformed YANG models by identifying alignments between key concepts in these models and those in a more expressive ontology.

As an example, the YANG-KG-SEMANTIC-GENERALIZATION case could involve wanting to integrate Service and Network topology data, matching the Network Topologies [RFC8345] and Service Assurance [RFC9418] YANG data models, into a knowledge graph structured by the NORIA-O ontology [NORIA-O-2024].

Although identifying alignments in the YANG-KG-SEMANTIC-GENERALIZATION case may appear non-trivial for "constructor" YANG models, it is worth noting that the design of YANG models generally relies on principles of concept hierarchies and reuse of common concepts between models to promote model interoperability, as is the case with the Abstract Network Model of [RFC8345]. Therefore, the task of identifying alignments can theoretically benefit from these design principles.

In continuity of the above RFC8345 / NORIA-O example, providing an alignment may mean asserting a semantic equivalence between the RDFS/OWL representation of the "node" concept from [RFC8345] with the "noria:Resource" concept from [NORIA-O-2024]. Examples of approaches for linking ontologies are provided in Section 4.2.

4.2. Implementing Alignments of Model-Specificities to a Multi-Faceted Knowledge Graph

Building on the previously defined YANG-KG-SEMANTIC-GENERALIZATION scenario, this section presents two approaches to construct the structuring ontology of the ITSM-KG by combining YANG models translated into RDFS/OWL and a meta-ontology enabling the analysis of the operational context of the network lifecycle. As techniques for identifying alignments between data models is beyond the scope of this document, we refer interested readers to specialized literature in this field, such as [ONTO-MATCH-2022].

To present the approaches, we assume the ability to convert a given YANG model into its ONTO-YANG-MODEL (i.e. its equivalent RDFS/OWL representation). The code snippet in Figure 2 is a fictional example of translating the "node" concept from [RFC8345] into its RDFS/OWL equivalent.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<urn:ietf:params:xml:ns:yang:ietf-network#node>
  rdf:type owl:Class ;
  rdfs:comment  "The inventory of nodes of this network." ;
.
```

Figure 2: Snippet of the ONTO-YANG-MODEL describing the ’node’ concept from RFC8345 into its RDFS/OWL equivalent, in Turtle syntax.

The following sub-sections build on the ONTO-YANG-MODEL example from Figure 2.

4.2.1. The Network of Ontologies Approach

The network of ontologies approach is a common practice in the field of knowledge engineering and Semantic Web technologies. The principle involves assembling vocabularies from different domains to form a coherent set, for example to infer - through graph traversal or reasoning - relationships between entities in the graph, starting from a concept defined in one of the vocabularies and leading to an instance of a concept from another vocabulary.

In our example, the code snippet of Figure 3 implements the ONTO-ITSM by importing concepts from the ONTO-YANG-MODEL (Figure 2) and concepts from the ONTO-META (Figure 4). An additional import in Figure 5 relates to the ONTO-LINKER.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<https://example.com/ontologies/itsm/>
  rdf:type owl:Ontology ;
  owl:imports
    # ===> Import of one of the ONTO-YANG-MODEL <===
    <https://example.com/ontologies/ietf-network-topology> ,
    # ===> Import of the ONTO-META <===
    <https://w3id.org/noria/ontology/> ,
    # ===> Import of the ONTO-LINKER definitions <===
    <https://example.com/ontologies/ietf-noria-linker> ;
.
```

Figure 3: The implementation of the ONTO-ITSM to structure the relation of ONTO-YANG-MODEL(s) with ONTO-META, in Turtle syntax.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix seas: <https://w3id.org/seas/>.  # Smart Energy Aware Systems
@prefix bot:  <https://w3id.org/bot#> .  # Building Topology Ontology
@prefix observable:  # Unified Cybersecurity Ontology (UCO)
  <https://unifiedcyberontology.org/ontology/uco/observable#> .
@prefix log: <https://w3id.org/sepses/ns/log#> .  # a.k.a. SLOGERT

@prefix noria: <https://w3id.org/noria/ontology/> .

noria:Resource
    rdf:type owl:Class ;
    rdfs:label "Resource" ;
    rdfs:comment """General resource record of the Communication Device
      kind from the logistics park. It is a managed entity that can be
      either Physical or Virtual."""@en ;
    rdfs:subClassOf noria:StructuralElement ;
    rdfs:subClassOf
        seas:System,
        seas:CommunicationDevice,
        bot:Element ,
        observable:Device ,
        log:Host ;
    rdfs:isDefinedBy noria: ;
.
```

Figure 4: Snippet of the ONTO-META describing the 'noria:Resource' concept from NORIA-O v0.3, in Turtle syntax.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix noria: <https://w3id.org/noria/ontology/> .

noria:Resource
  owl:equivalentClass <urn:ietf:params:xml:ns:yang:ietf-network#node> ;
.
```

Figure 5: Snippet of the ONTO-LINKER to relate ONTO-YANG-MODEL definition(s) with ONTO-META definition(s), in Turtle syntax.

As a result, querying any ITSM-KG structured by the ONTO-ITSM, as shown in Figure 6, enables retrieving entities of the ITSM-KG using ONTO-META concepts, even if entities are described with ONTO-YANG-MODEL concepts.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX noria: <https://w3id.org/noria/ontology/>

SELECT ?res

WHERE {
  # Pattern for the base class from ONTO-META
  # or any equivalent class from ONTO-YANG-MODEL
  ?resClass (owl:equivalentClass|^owl:equivalentClass)* noria:Resource .

  # Pattern to retrieve instances from the ITSM-KG
  ?res rdf:type ?resClass .
}
```

Figure 6: Snippet to retrieve entities of the ITSM-KG assuming the relatedness of ONTO-META concepts with ONTO-YANG-MODEL concepts, in SPARQL syntax.

4.2.2. Explicit Linking in the ONTO-META

In this approach, we assume that we have the means to evolve ONTO-META, which allows for the implementation of equivalence relationships between the concepts of ONTO-META and ONTO-YANG-MODEL directly within ONTO-META, as shown in Figure 7.

In this sense, ONTO-ITSM is part of ONTO-META, and ONTO-LINKER is within ONTO-META. The query in Figure 6 applies here as well and will yield the same results.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix seas: <https://w3id.org/seas/>.  # Smart Energy Aware Systems
@prefix bot:  <https://w3id.org/bot#> .  # Building Topology Ontology
@prefix observable:  # Unified Cybersecurity Ontology (UCO)
  <https://unifiedcyberontology.org/ontology/uco/observable#> .
@prefix log: <https://w3id.org/sepses/ns/log#> .  # a.k.a. SLOGERT

@prefix noria: <https://w3id.org/noria/ontology/> .

<https://w3id.org/noria/ontology/>
  a owl:Ontology ;
  # ===> Import of one of the ONTO-YANG-MODEL <===
  <https://example.com/ontologies/ietf-network-topology> .

noria:Resource
    rdf:type owl:Class ;
    rdfs:label "Resource" ;
    rdfs:comment """General resource record of the Communication Device
      kind from the logistics park. It is a managed entity that can be
      either Physical or Virtual."""@en ;
    rdfs:subClassOf noria:StructuralElement ;
    rdfs:subClassOf
        seas:System,
        seas:CommunicationDevice,
        bot:Element ,
        observable:Device ,
        log:Host ;
    rdfs:isDefinedBy noria: ;
    # ===> Explicit linking to ONTO-YANG-MODEL <===
    owl:equivalentClass <urn:ietf:params:xml:ns:yang:ietf-network#node>
.
```

Figure 7: Snippet of the ONTO-META describing the 'noria:Resource' concept from NORIA-O v0.3 with added linking to ONTO-YANG-MODEL, in Turtle syntax.

4.3. Extract-Transform-Load Pipelines for the ITSM-KG

Based on [I-D.marcas-nmop-knowledge-graph-yang] and [NORIA-DI-2023], which present the technical means to implement a pipeline for constructing the ITSM-KG, this section focuses on two complementary viewpoints: Section 4.3.1 the management of streaming data such as alarms and logs, and Section 4.3.2 the deployment of a federated data architecture when various technical foundations or business units are involved in providing the ITSM-KG.

From the perspective of the Digital Map Requirements (Section 3.2), the Figure 10, Figure 11 and Figure 12 particularly address the REQ-DM-SCALES requirement.

4.3.1. Handling Event Streams

The following figures illustrate different scenarios for constructing a ITSM-KG through an Extract-Transform-Load (ETL) data integration pipeline.

Figure 8 illustrates a common design pattern providing the capability to record event streams into a knowledge graph, such as an ITMS-KG if considering that event data are mapped to ONTO-META concepts and network entities to ONTO-YANG-MODEL concepts. The Figure 9 provides an example of the resulting representation in the form of a knowledge graph.

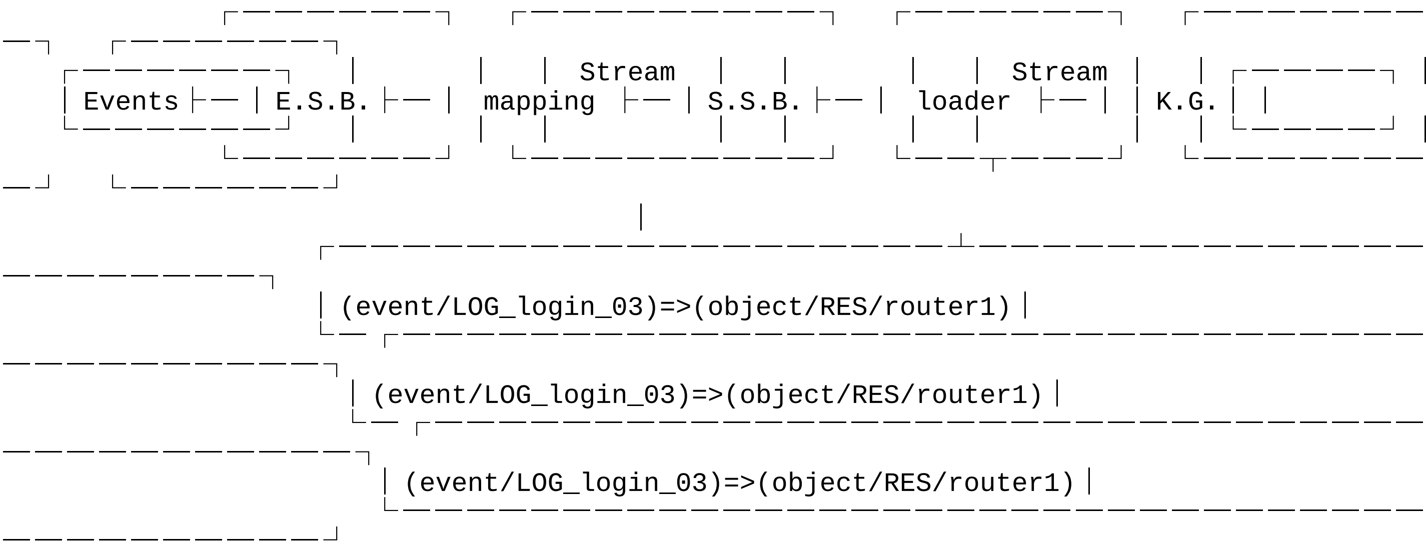

Figure 8: KG-only data integration architecture for event data streams.

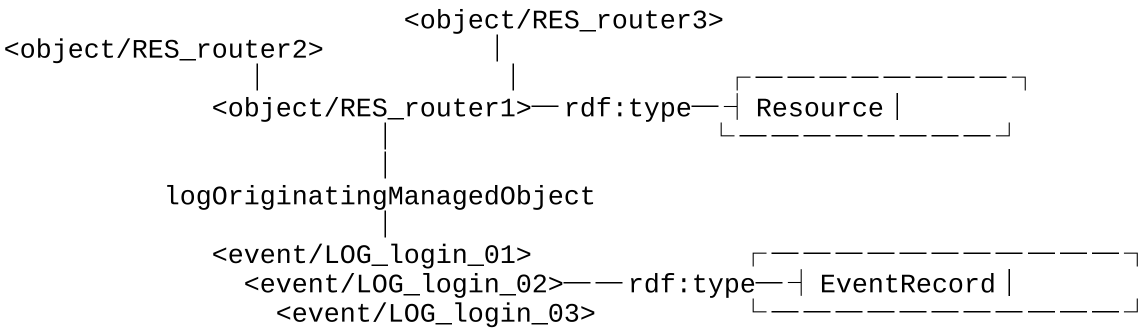

Figure 9: Resulting knowledge representation for the KG-only data integration architecture for event data streams

As event streams can be high-paced, it could be beneficial to leverage input/output (I/O) performance optimizations specific to each type of database management system (DBMS), such as Time-Series DataBases (TSDBs) for streaming data and graph databases for knowledge graphs. Figure 10 illustrates the capability to handle both a knowledge graph and a time-series representation of the network’s lifecycle while maintaining a link between the two representations (Figure 11). Each serve different purposes, such as context analysis with the knowledge graph representation and trend analysis with the TSDB. Thanks to the linking between the two storage systems, users browsing aggregated data from the knowledge graph can access the raw data within the relevant time span for further analysis, and vice versa.

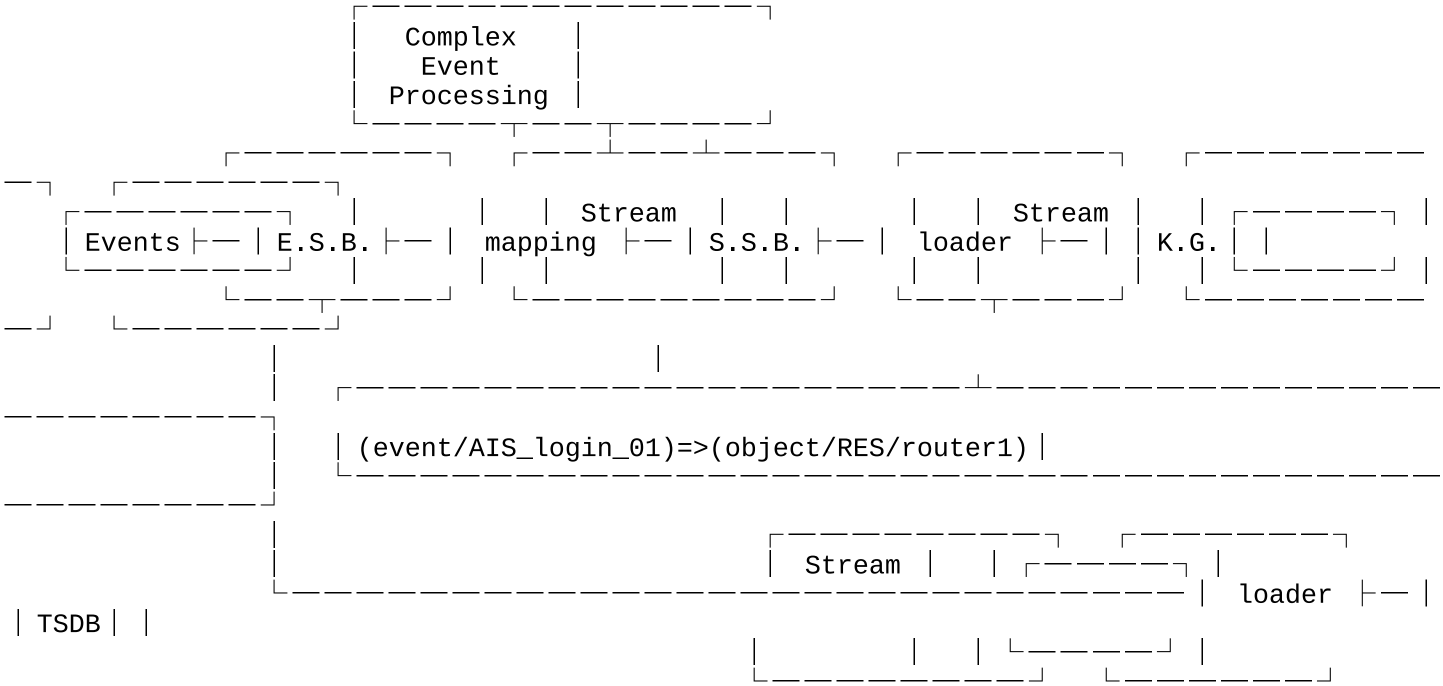

Figure 10: Mixed KG/non-KG data integration architecture for event data streams.

```
                                   <object/RES_router3>
          <object/RES_router2>           │
                          │               │             ┌─────────────────┐
                       <object/RES_router1>──rdf:type──┤ Resource │
                                  │                    └─────────────────┘
                    logOriginatingManagedObject
                                  │                     ┌────────────────────────┐
    ┌────────────────────────────────────<event/AIS_login_01>──rdf:type──┤ EventRecord │
    │                     │              │   \            └────────────────────────┘
    │                duration           │    \
    │                     │              │  dcterms:type
    │  "P0Y0M0DT0H3M30S"^^xsd:duration  │      \
    │                                   │    <Notification/
    │                         loggingTime   EventType/inferredAlert>
    │                                   │                    │
    │      "2024-02-07T16:22:42Z"^^xsd:dateTime       rdf:type
    │                                                ┌───────────┴──────────┐
    │                                                │ skos:Concept │
    │  KG knowledge representation                   └──────────────────────┘
    │  ==============================================================
    │  Time series database (TSDB) data representation
    │
    │  Timestamp             Origin                Event
    │  2024-02-07T16:22:42Z  <object/RES_router1>  Login Attempt
    │  2024-02-07T16:23:13Z  <object/RES_router1>  Login Attempt
    │  2024-02-07T16:26:12Z  <object/RES_router1>  Login Attempt
    │                                  ▲
    └───shared──identifier─────────────────────────────────┘
```

Figure 11: Resulting knowledge representation for the mixed KG/ non-KG data integration architecture for event data streams.

4.3.2. Federated Data Architecture

The Figure 12 illustrates the principles for providing unified access to data distributed across various technological platforms and stakeholders thanks to Federated Queries [SPARQL11-FQ] and the use of a shared ONTO-ITSM across data management platforms.

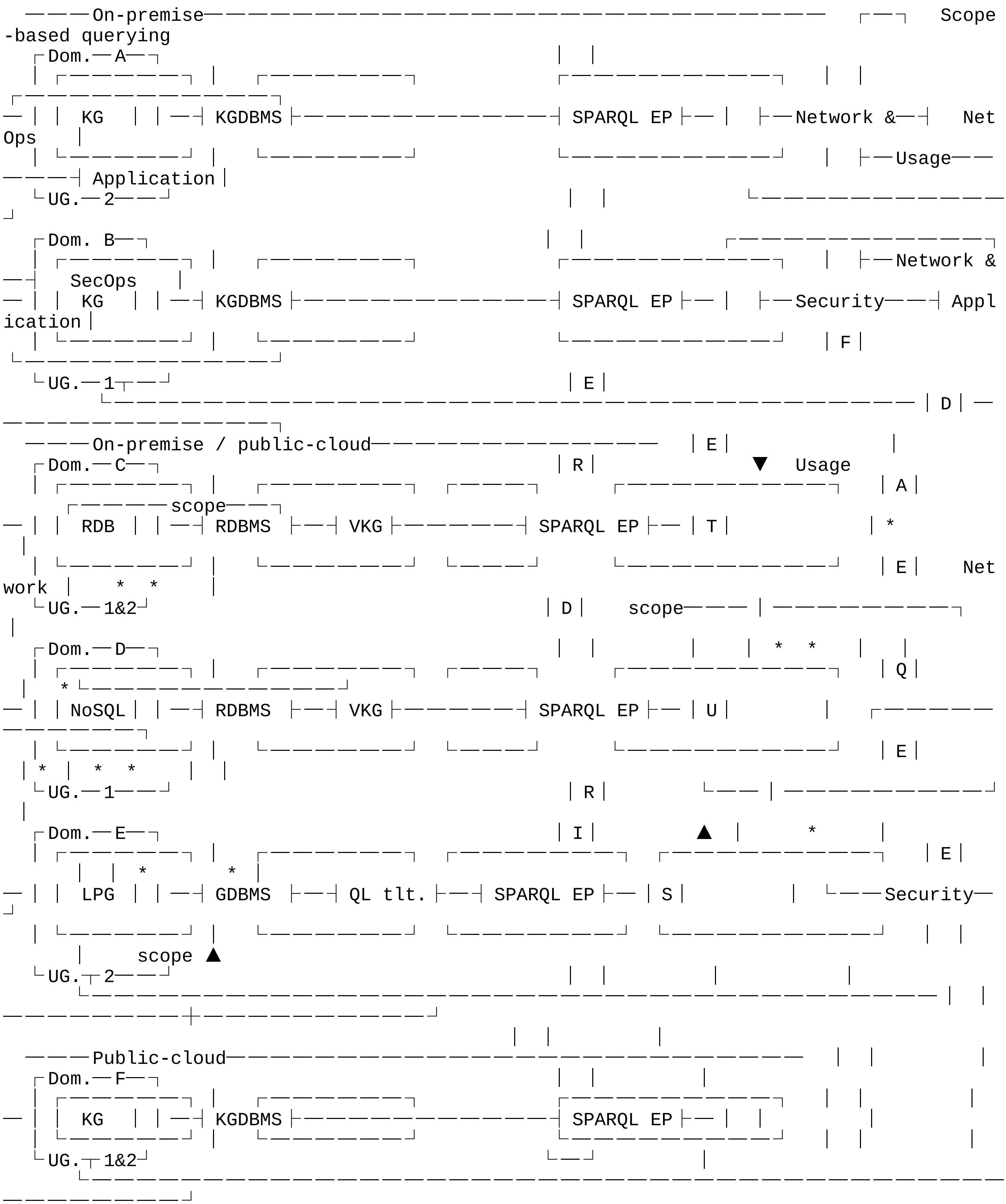

Figure 12: Unified access to data distributed across various technological platforms.

5. Experiments

5.1. Experimental Plan

In terms of experimentation, we consider the YANG-KG-SEMANTIC-GENERALIZATION case defined in Section 4 as the reference approach and recommend implementing a data processing pipeline that performs the following use cases:

Y-MODEL-FROM-DATA: Based on a dataset of configuration data expressed in YANG models, the goal is to enable extracting the list of models involved for their conversion to their RDFS/OWL equivalent.

Y-MODEL-DEPENDENCIES: Based on a given YANG model, the goal is to enable identifying and retrieving all the YANG models that the model refers to, in order to build a complete corpus of models for their conversion to their RDFS/OWL equivalent as a coherent set.

Y-MODEL-TO-RDFS-OWL: Based on a YANG model and the associated model corpus (i.e. Y-MODEL-DEPENDENCIES), the goal is to enable producing a semantically equivalent RDFS/OWL representation (i.e. ONTO-YANG-MODEL).

Ideally, a YANG to RDFS/OWL/YANG projection algebra would be used to provide a formal proof of semantic equivalence; testing mechanisms should be implemented as a fallback to provide a proof of equivalence.

Y-INSTANCE-TO-KG: Based on a dataset of configuration data expressed in YANG models and the related (set of) ONTO-YANG-MODEL, the goal is to enable constructing a knowledge graph from the configuration data, with the knowledge graph structured by the (set of) ONTO-YANG-MODEL.

Y-MODEL-META-KG-ALIGNMENT: Based on a corpus of YANG models transformed into RDFS/OWL (i.e. Y-MODEL-TO-RDFS-OWL) and a reference ontology structuring the ITSM-KG, the goal is to enable querying of the configuration entities present in the graph (i.e. data derived from the Y-INSTANCE-TO-KG case) through the concepts of the reference ontology.

In addition to identifying the class and property correspondences between the resulting Y-MODEL-TO-RDFS-OWL models and the reference ontology, this capability requires implementing a necessary and sufficient number of class equivalence relations and property equivalence relations.

META-KG-BEHAVIORAL-MODEL: Based on the ITSM-KG, which results from the composition of the Y-INSTANCE-TO-KG case with Y-MODEL-META-KG-ALIGNMENT and additional operational data structured by ONTO-META, the goal is to learn behavioral models (e.g. incident signatures) in a formalism that can be interpreted through the lenses of ONTO-ITSM and shared with other stakeholders with minimal discrepancies in the underlying configuration data.

5.2. Implementation Status

This section provides pointers to existing open source implementations of this document or in close relation to it.

5.2.1. NORIA

The NORIA project aims at enabling advanced network anomaly detection using knowledge graphs. Among the components resulting from this project, the following ones serve the use case described in this document:

* NORIA-O [NORIA-O-2024], is a data model for IT networks, events and operations information. The ontology is developed using web technologies (e.g. RDF, OWL, SKOS) and is intended as a structure for realizing an ITSM knowledge graph for Anomaly Detection (AD) and Risk Management applications. The NORIA-O implementation is available as open source at https://w3id.org/noria/ (https://w3id.org/noria/). Its use for anomaly detection is discussed in:

  - [SLKG-2023] with a model-based design approach (i.e. query the graph to retrieve anomalies and their context) and a statistical learning approach (i.e. relate entities based on context similarities, then use this relatedness to alert and guide the repair).

  - [GPL-2024] with a process mining approach to align a sequence of entities to activity models, then use this relatedness to guide the repair actions.

  - [NORIA-UI-2024] a Web-based knowledge graph exploration design for incident management that combines the above [SLKG-2023] and [GPL-2024] techniques for broader coverage of anomaly cases and knowledge capitalization.

* A knowledge graph-based platform design [NORIA-DI-2023] using Semantic Web technologies and open source data integration tools to build an ITSM knowledge graph:

  - SMASSIF-RML, a Semantic Web stream processing solution with declarative data mapping capability. Available as open source at https://github.com/Orange-OpenSource/smassif-rml (https://github.com/Orange-OpenSource/smassif-rml).

  - ssb-consum-up, a Kafka to SPARQL gateway enabling end-to-end Semantic Web data flow architecture with a Semantic Service Bus (SSB) approach. Available as open source at https://github.com/Orange-OpenSource/ssb-consum-up (https://github.com/Orange-OpenSource/ssb-consum-up).

  - grlc, a fork of CLARIAH/grlc with SPARQL UPDATE and GitLab interface features to facilitate the call and versioning of stored user queries in SPARQL syntax (e.g. for anomaly detection following the model-based design approach). Available as open source at https://github.com/Orange-OpenSource/grlc (https://github.com/Orange-OpenSource/grlc).

* SemNIDS [SemNIDS-2023], a test bench involving network trafic generation, open source Network Intrusion Detection Systems (NIDS), knowledge graphs, process mining and conformance checking components.

Note that the NORIA project does not currently address the Y-MODEL-FROM-DATA, Y-MODEL-DEPENDENCIES, and Y-MODEL-TO-RDFS-OWL use cases.

5.2.2. YANG2OWL

The YANG2OWL framework aims at facilitating the implementation of a Network Digital Twin (NDT) that would leverage the representation and reasoning capabilities typically associated with knowledge graphs for anomaly detection needs, as well as for network management purposes by enabling network configuration based on modifications at the level of the ITSM-KG itself. Basically, the approach consists of reusing YANG data models used in network operations in a nearly equivalent form within Semantic Web technologies (i.e. producing ONTO-YANG-MODEL instances) to create a bijection between network configuration data and the NDT.

The YANG2OWL framework addresses the use cases Y-MODEL-TO-RDFS-OWL and Y-INSTANCE-TO-KG (as defined in Section 5.1).

Figure 13 illustrates the top-level tasks of the semantization process at play. Subsequent sections detail how the framework builds ontologies that captures the specificities of the telco domain and models any telco network instance as an ITSM-KG. Please note that the publication of the related tools and algorithms is in progress.

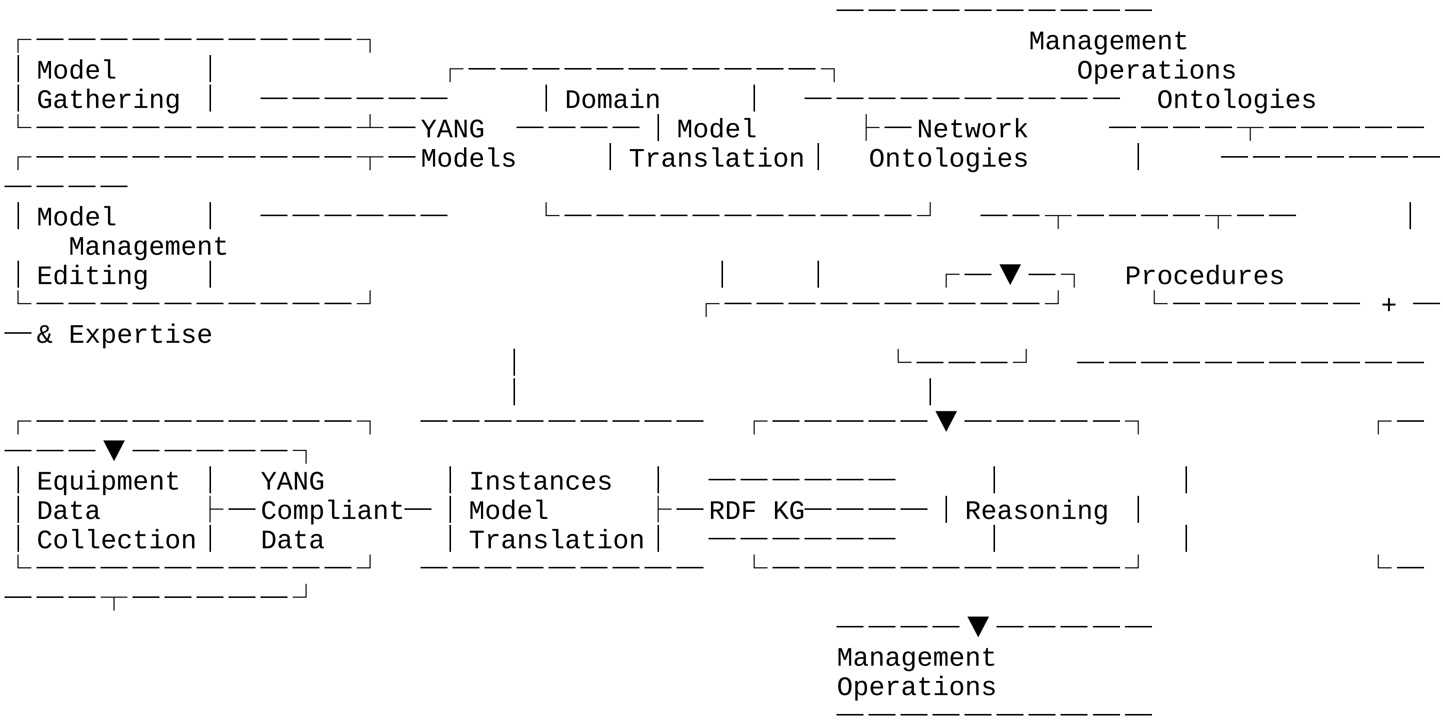

Figure 13: The YANG2OWL framework. Labels within boxes represent automated or human actions, while labels between top/bottom lines represent datasets

5.2.2.1. Motivations and Principles

The document [I-D.mackey-nmop-kg-for-netops] (Knowledge Graph Framework for Network Operations) emphasizes the importance of ontologies alongside knowledge graphs for network management automation. However, it lacks guidance on creating these ontologies and provides limited details on generating knowledge graphs or their relationship with the ontologies. To address these topics, the following principles have been considered to underpin the development of the YANG2OWL approach:

1. The ontologies should intimately reflect YANG models,

2. The generation of ontologies should be mostly automatized,

3. The knowledge graphs should intimately reflect the payload of messages that YANG compliant network equipments and controlers publish or emit in response to a Remote Procedure Call (RPC) request,

4. The generation of knowledge graphs should be automated,

5. The nodes and predicates of the knowledge graphs should be defined as instances of classes and properties of the ontologies.

Point 1 of the proposed principles is essential for ensuring the engagement of network administrators and experts in semantic technology. Aligning the ontology’s vocabulary (class and relationship naming) and semantics (relationship constraints) with that of network managers is crucial. The YANG language is currently the reference in this area and will continue to be so, given its specification by the IETF and support from major telco industry players. This necessity has driven the development of the YANG2OWL framework for converting YANG models into OWL models, which corresponds to point 2 of the proposal. Points 3, 4, and 5 are direct outcomes of the commitment to points 1 and 2.

5.2.2.2. The Y-MODEL-TO-RDFS-OWL step

YANG and OWL are both data modeling languages. They define a vocabulary and a grammar. The vocabulary defines the concept of the domain. YANG domain is the telco domain.

In a natural language, the vocabulary defines nouns, verbs, adjectives, and adverbs that are useful for discussing the world. The grammar specifies how these elements should be assembled into sentences that describe a state of the world. In a YANG model, the vocabulary is defined in terms of _containers_, _lists_, _leaves_, _leaf-lists_, and other categories, while the grammar is defined in terms of statements that relate these elements to one another. In an OWL ontology, the vocabulary is defined in terms of _classes_, _subclasses_, _object properties_, and _data properties_, which is somewhat similar to YANG but does not directly map.

As ontologies have been introduced as a modeling language meant to share a common view (or knowledge) of a domain among different stakeholders [GRUBER-1995], the terms defined by the ontologies should reflect those used by equipment manufacturers, telecom solutions developers, systems integrators, network operators, and ultimately end users.

A YANG model is a document containing declarations. The document has a tree-like structure: declarations can contain other declarations. There are about half hundred types of declarations. The main ones are _container_, _list_, _leaf_ and _leaf-list_:

CONTAINER: It is a concept, something we can talk about ; it is the the basic type of elements of the domain, such as a network, a node, a link. A container declaration can contain another container declaration that can be called a sub-container. This sub-container allows to define a concept that will characterize the container that contains it (e.g. link, source, and destination).

LIST: It is a concept that can have multiple instances, such as nodes of a network.

LEAF: It is a property of this concept, such as an identifier or a geographical location.

LEAF-LIST: It is a multivalued property, such as hours of the day the device is in sleep mode.

By applying the above principles, and in line with the reasons sketched in Section 5.2.2.1, we have developed the YANG2OWL that automatically generates OWL ontologies from YANG modules (i.e. computes ONTO-YANG-MODELs). Figure 14 sketches the use of the YANG2OWL tool to compute the org.opendaylight.yangtools ONTO-YANG-MODEL.

```
       YANG file
           |
           ▼
org.opendaylight.yangtools────Abstract Syntax Tree
                                     |
                                     ▼
       IETF RFC 7950──────────Yang2OwlConverter────OWL file
```

Figure 14: Computing the org.opendaylight.yangtools ONTO-YANG-MODEL with YANG2OWL.

In more detail, we have defined mapping rules between YANG constructs and OWL concepts and implemented these in YANG2OWL. The main YANG constructs (_container_, _list_, _leaf_, and _leaf-list_) are transformed as follows:

* The _container_ and _list_ declarations are converted into OWL classes. The name of the OWL class correponds to the name of the _container_ or _list_ in the YANG model.

* The _leaf_ and _leaf-list_ declarations are converted into OWL data properties. The name of the OWL data property corresponds the name of the _leaf_ or _leaf-list_ in the YANG model.

An example of this conversion is presented in the following section.

5.2.2.3. The Y-INSTANCE-TO-KG step

As introduced above, YANG models define the vocabulary and grammar to describe factual knowledge about the state of the network. For example if a YANG module defines the container _node_, and this container has a leaf identifier which has the type string, then a valid JSON document with configuration data describing a node should be a JSON object containing a key named identifier which value should be a string such as router_253.

So, in line with the mapping rules of YANG statement into OWL concepts defined in Section 5.2.2.2, when parsing a JSON tree that comply to a given YANG model we can assume that if we get a _key which value is a JSON object_ then the _key should be the name of a container or a list_ and its _value should be a description to be further analyzed_. Thus, in terms of knowledge graph modeling, this JSON object should be interpreted as an _instance of a class_ which name is the _name of the container or of the list_.

Conversely, if the value is a _litteral_, the _key_ should be the _name of a leaf or a leaf-list_. Thus, in terms of knowledge graph modeling, the litteral should be interpreted as the _object of a DataProperty_ which name is the _name of the leaf_.

The JSON2RDF tool (which is part of the YANG2OWL framework) implements these principles, realizing the Y-INSTANCE-TO-KG use case. Figure 15 shows the algorithm implemented by JSON2RDF as pseudo code.

```
function createURI(jsonObject, class, namespace, ontology) {
  if class has a 'key' annotation {
    get the content <keycontent> of this annotation
    search the key <keycontent> in the jsonObject
    append the key to the namespace to create the URI
  } else {
    generate a unique URI
  }
  return the URI created
}

function createObject(URI, class) {
  return an instance of the class with the given URI
}

function parse(object, parentURI, class, namespace, ontology) {
  objectURI = createURI(object,class, namespace, ontology)
  createObject(objectURI, class)
  for each key of object {
    if the value of object[key] is a list {
      for each elt of the list {
        if elt is an object {
          parse(elt, objectURI, key, namespace, ontology)
          create the triple <objectURI haskey elt>
        } else if elt is a literal
            create the triple <objectURI key elt>
    } else if the value of object[key] is an object {
        eltURI = createURI(elt,key, namespace, ontology)
        create the triple <objectURI haskey eltURI>
        parse(elt, objectURI, key, namespace, ontology)
    } else if the value of object[key] is literal {
        create the triple <objectURI key value>
    }
  }
}
```

Figure 15: Pseudo code of the algorithm implemented by JSON2RDF.

The algorithm is initiated by calling the parse function as follows, where top is the root of the JSON object (i.e. configuration data as a JSON tree that complies to a given YANG model), and ontology is the output of the Y-MODEL-TO-RDFS-OWL step:

```
call parse(top, nil, namespace, ontology)
```

5.2.2.4. Example of Implementation

To illustrate the YANG2OWL approach, this section briefly reports on an experiment conducted in an industrial setting with data from a virtualized 5G infrastructure. In the context of the Network Change Management process, _impact analysis_ prior to conducting a scheduled operation can be run on an ITSM-KG. It aims to determine all the components of the 5G core network that are dependent of a given (set of) network infrastructure element. For example, for a scheduled operation on a leaf node (i.e. a network element in a 2-tier spine-leaf architecture), the impact calculus will return all the servers connected to the leaf, all the Virtual Machines (VMs) hosted on these servers, all the Network Functions (NFs) deployed on these VMs, and ideally all the telecom services using these NFs.

Figure 16 provides an overview of the data processing workflow used for the experiment. The tasks of the diagram are described below.

```
                  START
           ┌───────────────┘   └───────────────┐
           ▼                   │
    Model                     │
    Gathering                 │
           │                   │
           ▼                   │
    │ Model                    │
    │ Translation              │
           │                   │
           ▼                   │
    Model                     │
    Curation                  │
           │                   │
           ▼                   ▼
    │ Model-Related        │ NetOps-Related
    │ Knowledge Graph      │ Knowledge Graph
    │ Construction         │ Construction
           │                   │
           └───────────────┐   ┌───────────────┘
                        ▼  ▼
             │ Global
             │ Knowledge Graph
             │ Construction
                    │
                    ▼
             │ Use Cases-Related
             │ Pre-Processing
                    │
                    ▼
             │ Use Cases-Related
             │ Querying
                    │
                    ▼
              Situation
              Analysis
                    │
                    ▼
                   END
```

Figure 16: Flowchart for the YANG2OWL experiment. A left vertical bar on a step indicates that it is scripted; otherwise, steps require user or operator action.

Model Gathering: This task corresponds to the realization of the Y-MODEL-FROM-DATA use case with the manual selection of YANG modules in relation to the 3GPP application domain. The YANG modules from [ETSI-TS-128-541] have been selected for this experiment.

Model Translation: For a given YANG module, this task implements the Y-MODEL-DEPENDENCIES use case by fetching sub-YANG modules from well-known GitHub repositories used for storing YANG modules (e.g. IETF, IEEE, IANA, ETSI, broadband forum, OpenROADM, OpenConfig, Cisco, Huawei, to name a few). This is achieved by scrutinizing import clauses (including imports of imports) and examining module locations and relationships from the [YANG-CATALOG]. Additionally, it addresses the Y-MODEL-TO-RDFS-OWL use case using the YANG2OWL solution defined in Section 5.2.2.2. For this experiment, the resulting ontology is referred to as MOBILE-O.

Model Curation: This task involves providing a streamlined ontology by manually _filtering_ (selection of classes and relationships based on the data available) and _grouping_ (compression of the model hierarchy, i.e. class of classes) the model resulting from the _Model Translation_ task. This simplification aims to enhance the readability of the model for an operator and facilitate the implementation of potentially more concise queries in the downstream _Use Cases-Related Querying_ task.

Model-Related Knowledge Graph Construction: It realizes the Y-INSTANCE-TO-KG use case using the JSON2RDF solution described in Section 5.2.2.3.

NetOps-Related Knowledge Graph Construction: It corresponds to the execution of RML transformation rules [RML] with definitions from the NORIA-O ontology [NORIA-O-2024] for the integration of complementary data to that of the 5G network derived from YANG configurations (i.e. the _Model-Related Knowledge Graph Construction_ task), such as the topology of connected networks, scheduled operations, incident tickets, and organization-related data.

Global Knowledge Graph Construction: It is achieved through parallel insertions into a graph database of the results from the _Model-Related_ and _NetOps-Related_ tasks, after ensuring that: 1) the URI patterns implemented in the RML rules of the NetOps-Related step are consistent with the URIs produced by the Model-Related step to benefit from automatic linking of triples within the graph database through the uniqueness of the URIs; 2) the definition of mappings between MOBILE-O and NORIA-O has been implemented and inserted into the graph database (i.e. realization of the Y-MODEL-META-KG-ALIGNMENT use case through the implementation of the ONTO-LINKER concept as illustrated in Figure 5). For this experiment, the graph database is a Neo4j database [NEO4J] instance, and the loading is performed using the Neo4j Neosemantics toolkit.

Use Cases-Related Pre-Processing: Dependency relationships are, in general, knowledge elements that cannot be directly derived from field data; they are part of the business knowledge regarding the operation of the network systems. It may therefore be beneficial to support the downstream _Use Cases-Related Querying_ task by performing pre-processing, particularly by calculating these dependency relationships retrospectively from business rules and the data loaded into the database. For example, one can create a (Server)-[DEPENDS_ON]->(Leaf) relationship by searching instances of the (Server)-(Server Interface)-(Network Link)-(Leaf Interface)-(Leaf) graph pattern. The same principle can apply to different network configurations to create other kinds of dependency relationships.

For this experiment, the dependency relationships are calculated directly in the graph database using Neo4j Cypher language queries, or externally to the graph database using SHACL shapes [SHACL] according to the principles described in [GUITTOUM-2023]. As another example, more specific to the 3GPP models [ETSI-TS-128-541] included in MOBILE-O and the Neo4j setup, one could calculate a dependency relationship between a 5G NF and the Kubernetes cluster that hosts it, as shown in Figure 17. It is important to note that subclass inference with Neo4j is not automatic and must be performed through dedicated queries, as illustrated in Figure 18.

```
MATCH (c:ManagedFunction)--(n:namespace)--(k:ClusterKubernetes)
MERGE (c)-[d:DEPENDS_ON]->(k)
```

Figure 17: Dependency calculation query, in Cypher syntax, for relating a 5G NF and the Kubernetes cluster that hosts it.

```
MATCH (m)<-[:subClassOf]-(x)<-[:type]-(c)
WHERE m.uri CONTAINS 'ManagedFunction'
SET c:ManagedFunction
```

Figure 18: Subclass inference query, in Cypher syntax, to tag 5G NF entities as ‘ManagedFunction‘ based on prior annotation of the entities at creation time with a specific class described in the YANG model, which is also a subclass of ‘ManagedFunction‘ as per MOBILE-O.

Use Cases-Related Querying: The exploitation of dependency relationships is carried out through queries on the graph, e.g. during the insertion of an entity of type noria:ChangeRequest or by following an exploratory approach by coupling a query such as that in Figure 19 with a visualization tool like Neo4j NeoDash.

```
MATCH (e1) WHERE e1.resourceHostName = $neodash_ressource_hostname
MATCH q1 = (e1) ((w)<-[:DEPENDS_ON]-(t)) {0,8}
UNWIND t AS impacts
RETURN DISTINCT impacts.resourceHostName
```

Figure 19: User query, in Cypher syntax using a quantified path pattern, for rendering dependency relationships in a Neo4j NeoDash display. The query seeks paths starting from the node ‘e1‘ and propagates up to 8 times using the ‘DEPENDS_ON‘ relationships. The depth of 8 has been defined in relation to the characteristics of the networks addressed in the experimentation.

Situation Analysis: Decision-making based on the results of the upstream task is the responsibility of the network administrator, potentially supported by a complementary exploration of the ITSM-KG performed algorithmically or interactively to analyze a broader technical and operational context.

5.2.2.5. Discussion

While the YANG2OWL approach has proven its validity as a proof of concept, several R&D questions remain for exploration with the NMOP community, including:

* Are the conversion principles based on statement types (class vs. data property) in the Y-MODEL-TO-RDFS-OWL use case universally applicable?

* How to ensure that an ITSM-KG can still be generically constructed from JSON/YANG data and queried when a _Model Curation_ task is applied on an ONTO-YANG-MODEL?

* What techniques can automate the Y-MODEL-META-KG-ALIGNMENT use case?

* What principles should guide the implementation of the Y-MODEL-META-KG-ALIGNMENT use case to extract an aggregated view from ONTO-META of infrastructures/configurations represented by an ONTO-YANG-MODEL (e.g. distinguishing devices from sub-devices)?

* As evoked in [I-D.boucadair-nmop-rfc3535-20years-later] (NEW-OPS-REQ-QUICK-BUT-WELL), how can we ensure reliable retrieval of dependencies between YANG modules for the Y-MODEL-DEPENDENCIES use case? Indeed, while browsing the GitHub projects of module developers, we observe a lack of uniformity in the way modules are presented and managed (e.g. differences in project structure, replication and local modifications of reference modules), which hinders dependency calculation and the sound inclusion of sub-modules in the YANG2OWL translation process.

Furthermore, it is noteworthy that the YANG2OWL approach is complementary to the YANG2RDF approach [YANG2RDF-IETF-121], which consists in translating YANG models into RDF. More specifically, YANG2RDF defines an ontology of the YANG language, where RDF graph instances model a YANG module. This approach is useful for querying YANG models. In contrast, the YANG2OWL approach defines an ontology of a YANG model, where RDF graph instances model an operational network. Future work may aim to combine the YANG2RDF and YANG2OWL approaches.

Finally, it is noteworthy that the YANG2OWL framework automates the _Ontology Implementation_ and _Ontology Update_ activities of the LOT4KG methodology [LOT4KG-2024] (a methodology that extends the well-known LOT ontology engineering methodology to include knowledge graph lifecycle management) by linking YANG modules with ITSM-KG fragment construction. This streamlines the development of NDT architectures based on knowledge graphs and simplifies ITSM-KG updates when YANG modules change.

6. Security Considerations

As this document covers the _ITSM-KG_ concepts, and use cases, there is no specific security considerations.

However, as the concept of a meta-knowledge graph involves the construction of a multi-faceted graph (i.e. including network topologies, operational data, and service and client data), it poses the risk of simplifying access to network operational data and functions that fall outside the knowledge graph users’ responsibility or that could facilitate the intervention of malicious individuals. To support the discussion on mitigating this risk, we suggest referring to Figure 12, which illustrates the concept of partial access to the meta-knowledge graph based on rights associated with each user group (UG) at the data domain level. We also recommend referring to [AMO-2012] for an example of implementation of access rights in a content management system that relies on Semantic Web models and technologies. This implementation uses the AMO ontology, which includes a set of classes and properties for annotating resources that require access control, as well as a base of inference rules that model the access management strategy to carry out.

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <https://www.rfc-editor.org/rfc/rfc8345>.

[RFC9418] Claise, B., Quilbeuf, J., Lucente, P., Fasano, P., and T. Arumugam, "A YANG Data Model for Service Assurance", RFC 9418, DOI 10.17487/RFC9418, July 2023, <https://www.rfc-editor.org/rfc/rfc9418>.

8.2. Informative References

[AMO-2012] Buffa, M. and C. Faron-Zucker, "Ontology-Based Access Rights Management", 2012, <https://doi.org/10.1007/978-3-642-25838-1_3>.

[DevOpsInfra-2021]
Corcho, O., Chaves-Fraga, D., Toledo, J., Arenas-Guerrero, J., Badenes-Olmedo, C., Wang, M., Peng, H., Burrett, N., Mora, J., and P. Zhang, "A High-Level Ontology Network for ICT Infrastructures", 2021, <https://doi.org/10.1007/978-3-030-88361-4_26>.

[ETSI-TS-128-541]
ETSI, "5G; Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (3GPP TS 28.541 version 18.9.0 Release 18)", October 2024, <https://www.etsi.org/deliver/etsi_ts/128500_128599/128541/18.09.00_60/ts_128541v180900p.pdf>.

[FLAGSM-2021]
Steenwinckel, B., Paepe, D. D., Hautte, S. V., Heyvaert, P., Bentefrit, M., Moens, P., Dimou, A., Bossche, B. V. D., Turck, F. D., Hoecke, S. V., and F. Ongenae, "FLAGS: A Methodology for Adaptive Anomaly Detection and Root Cause Analysis on Sensor Data Streams by Fusing Expert Knowledge with Machine Learning", 2021, <https://doi.org/10.1016/j.future.2020.10.015>.

[FOLIO-2018]
Steenwinckel, B., Heyvaert, P., Paepe, D. D., Janssens, O., Hautte, S. V., Dimou, A., Turck, F. D., Hoecke, S. V., and F. Ongenae, "Towards Adaptive Anomaly Detection and Root Cause Analysis by Automated Extraction of Knowledge from Risk Analyses", 2018, <https://www.ceur-ws.org/Vol-2213/paper2.pdf>.

[GPL-2024] Tailhardat, L., Stach, B., Chabot, Y., and R. Troncy, "Graphameleon: Relational Learning and Anomaly Detection on Web Navigation Traces Captured as Knowledge Graphs", 2024, <https://doi.org/10.1145/3589335.3651447>.

[GRUBER-1995]
R., G. T., "Toward principles for the design of ontologies used for knowledge sharing?", 1995, <https://doi.org/10.1006/ijhc.1995.1081>.

[GUITTOUM-2023]
Amal, G., Francois, A., Sbastien, B., Fabienne, B., and D. P. Noel, "Inferring Threatening IoT Dependencies Using Semantic Digital Twins Toward Collaborative IoT Device Management", 2023, <https://doi.org/10.1145/3555776.3578573>.

[I-D.boucadair-nmop-rfc3535-20years-later]
Boucadair, M., Contreras, L. M., de Dios, O. G., Graf, T., Rahman, R., and L. Tailhardat, "RFC 3535, 20 Years Later: An Update of Operators Requirements on Network Management Protocols and Modelling", Work in Progress, Internet-Draft, draft-boucadair-nmop-rfc3535-20years-later-08, 12 May 2025, <https://datatracker.ietf.org/doc/html/draft-boucadair-nmop-rfc3535-20years-later-08>.

[I-D.havel-nmop-digital-map-concept]
Havel, O., Claise, B., de Dios, O. G., and T. Graf, "Digital Map: Concept, Requirements, and Use Cases", Work in Progress, Internet-Draft, draft-havel-nmop-digital-map-concept-00, 4 July 2024, <https://datatracker.ietf.org/doc/html/draft-havel-nmop-digital-map-concept-00>.

[I-D.irtf-nmrg-network-digital-twin-arch]
Zhou, C., Yang, H., Duan, X., Lopez, D., Pastor, A., Wu, Q., Boucadair, M., and C. Jacquenet, "Network Digital Twin: Concepts and Reference Architecture", Work in Progress, Internet-Draft, draft-irtf-nmrg-network-digital-twin-arch-10, 28 February 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-network-digital-twin-arch-10>.

[I-D.mackey-nmop-kg-for-netops]
Mackey, M., Claise, B., Graf, T., Keller, H., Voyer, D., Lucente, P., and I. D. Martinez-Casanueva, "Knowledge Graph Framework for Network Operations", Work in Progress, Internet-Draft, draft-mackey-nmop-kg-for-netops-02, 4 March 2025, <https://datatracker.ietf.org/doc/html/draft-mackey-nmop-kg-for-netops-02>.

[I-D.marcas-nmop-knowledge-graph-yang]
Martinez-Casanueva, I. D., Rodrguez, L. C., and P. Martinez-Julia, "Knowledge Graphs for YANG-based Network Management", Work in Progress, Internet-Draft, draft-marcas-nmop-knowledge-graph-yang-05, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-marcas-nmop-knowledge-graph-yang-05>.

[I-D.netana-nmop-network-anomaly-lifecycle]
Riccobene, V., Roberto, A., Graf, T., Du, W., and A. H. Feng, "An Experiment: Network Anomaly Lifecycle", Work in Progress, Internet-Draft, draft-netana-nmop-network-anomaly-lifecycle-05, 3 November 2024, <https://datatracker.ietf.org/doc/html/draft-netana-nmop-network-anomaly-lifecycle-05>.

[LOT4KG-2024]
Romana, P., Mara, P.-V., Diego, C.-H., David, C.-F., and S. Lise, "When Ontologies Met Knowledge Graphs: Tale of a Methodology", 2024, <https://doi.org/10.1007/978-3-031-78952-6_43>.

[NEO4J] Neo4j, Inc., "Neo4j - Graph Database & Analytics", n.d., <https://neo4j.com/>.

[NORIA-DI-2023]
Tailhardat, L., Troncy, R., and Y. Chabot, "Designing NORIA: a Knowledge Graph-based Platform for Anomaly Detection and Incident Management in ICT Systems", 2023, <https://ceur-ws.org/Vol-3471/paper3.pdf>.

[NORIA-O-2024]
Tailhardat, L., Troncy, R., and Y. Chabot, "NORIA-O: An Ontology for Anomaly Detection and Incident Management in ICT Systems", 2024, <https://doi.org/10.1007/978-3-031-60635-9_2>.

[NORIA-UI-2024]
Tailhardat, L., Chabot, Y., Py, A., and P. Guillemette, "NORIA UI: Efficient Incident Management on Large-Scale ICT Systems Represented as Knowledge Graphs", 2024, <https://doi.org/10.1145/3664476.3670438>.

[ONTO-MATCH-2022]
Jan, P., Guilherme, C., Karolin, S., Katharina, K., Michael, H., and P. Heiko, "Ontology Matching Through Absolute Orientation of Embedding Spaces", 2022, <https://doi.org/10.1007/978-3-031-11609-4_29>.

[OWL] W3C, "OWL 2 Web Ontology Language Document Overview (Second Edition)", December 2012, <https://www.w3.org/TR/owl2-overview/>.

[RDF] W3C, "Resource Description Framework (RDF): Concepts and Abstract Syntax", February 2014, <https://www.w3.org/TR/rdf11-concepts/>.

[RDFS] W3C, "RDF Schema 1.1", February 2014, <https://www.w3.org/TR/rdf-schema/>.

[RML] Dimou, A., Sande, M. V., Meester, B. D., Heyvaert, P., and T. Delva, "RDF Mappling Language (RML)", June 2024, <https://rml.io/specs/rml/>.

[SemNIDS-2023] Ferrero, D., Agarwalla, Y., Tailhardat, L., and T. Ehrhart, "SemNIDS, bringing semantics into Network Intrusion Detection Systems", 2023, <https://github.com/D2KLab/SemNIDS>.

[SHACL] W3C, "Shapes Constraint Language (SHACL)", July 2017, <https://www.w3.org/TR/shacl/>.

[SKOS] W3C, "SKOS Simple Knowledge Organization System Reference", August 2009, <https://www.w3.org/TR/skos-reference/>.

[SLKG-2023] Tailhardat, L., Troncy, R., and Y. Chabot, "Leveraging Knowledge Graphs For Classifying Incident Situations in ICT Systems", 2023, <https://doi.org/10.1145/3600160.3604991>.

[SPARQL11-FQ] W3C, "SPARQL 1.1 Federated Query", March 2013, <https://www.w3.org/TR/sparql11-federated-query/>.

[SPARQL11-QL] W3C, "SPARQL 1.1 Query Language", March 2013, <https://www.w3.org/TR/sparql11-query/>.

[YANG-CATALOG] Cisco and IETF, "YANG Catalog", n.d., <https://www.yangcatalog.org/>.

[YANG2RDF-IETF-121] Michael, M., Anatolii, P., and C. Benoit, "YANG 2 RDF", 2024, <https://datatracker.ietf.org/doc/slides-121-nmop-yang-2-rdf/>.

Acknowledgments

We would like to thank Benoit Claise for spontaneously seeking to include the work of the NORIA research project in the vision of the NMOP working group through direct contact. We also extend our gratitude to Mohamed Boucadair for facilitating discussions within the NMOP community and for providing advice in organizing this Internet Draft.

Additionally, we would like to thank Fano Ramparany for his initial analysis of the possibilities of defining a model conversion algebra for going from YANG data models to OWL ontologies (draft-tailhardat-nmop-incident-management-noria-01).

Changes Between Revisions

v00 - v01 (draft-tailhardat-nmop-incident-management-noria)

* Added details to the _An ITSM-KG for Learning and Sharing Network Behavioral Models_ section (formerly called _A meta-knowledge graph to align operator-specificities and share behavioral models of technical architectures_).

* Added the Experiments / NORIA approach.

v01 - v02

* Added the Experiments / YANG2OWL framework based on details from Fano RAMPARANY (Orange Research), Pauline FOLZ (Orange Research), and Fabrice BLACHE (Orange Research).

* Added the Experiments / YANG2OWL example based on details from Romain VINEL (Orange France), Clment GOUILLOUD (SOFRECOM), Arij ELMAJED (Orange France), and Lionel TAILHARDAT (Orange Research).

Contributors

Maria Massri
Orange Research
Email: maria.massri@orange.com

Fabrice Blache
Orange Research
Email: fabrice.blache@orange.com

Sbastien Bolle
Orange Research
Email: sebastien.bolle@orange.com

Thomas Hassan
Orange Research
Email: thomas.hassan@orange.com

Romain Vinel
Orange France
Email: romain.vinel@orange.com

Arij Elmajed
Orange France
Email: arij.elmajed@orange.com

Clment Gouilloud
SOFRECOM
Email: clement.gouilloud@sofrecom.com

Authors' Addresses

Lionel Tailhardat
Orange Research
Email: lionel.tailhardat@orange.com

Raphal Troncy
EURECOM
Email: raphael.troncy@eurecom.fr

Yoan Chabot
Orange Research
Email: yoan.chabot@orange.com

Fano Ramparany
Orange Research
Email: fano.ramparany@orange.com

Pauline Folz
Orange Research
Email: pauline.folz@orange.com