Knowledge Graph Framework for Network Operations

Overview And Some Thoughts On Implementation

Knowledge Graph Framework for Network Operations

Knowledge Graph Framework for Network Operations

Knowledge Graph Framework for Network Operations

Knowledge Graph Framework for Network Operations

Knowledge Graph Framework for Network Operations

Knowledge Graph Framework for Network Operations

Knowledge Graph Framework for Network Operations

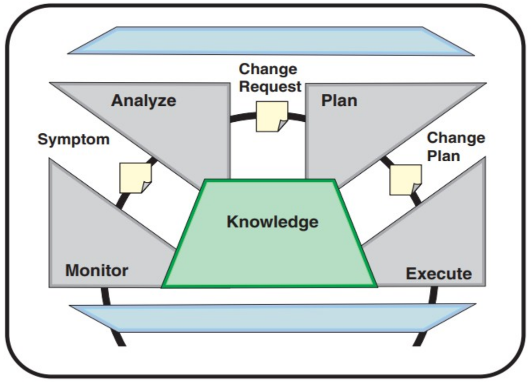
Knowledge Based Systems

Knowledge-based System: A knowledge-based system (KBS) is a computer program that reasons and uses a knowledge base to solve complex problems. The term is broad and refers to many different kinds of systems.

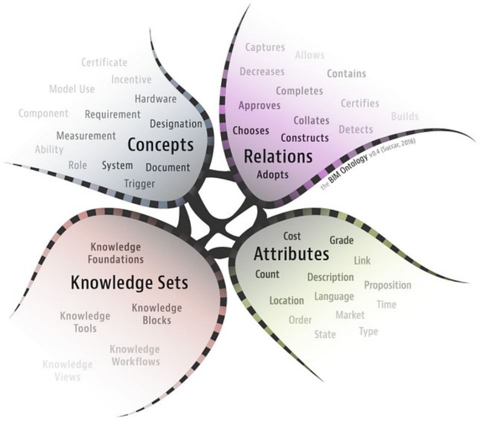
All knowledge based systems have two components:
A Knowledge Base: A way to represent knowledge explicitly.
An Inference Engine: a reasoning system that allows it to derive new knowledge.

A key way to capture facts about the world you are operating/observing is through the use of a Formal Ontology.

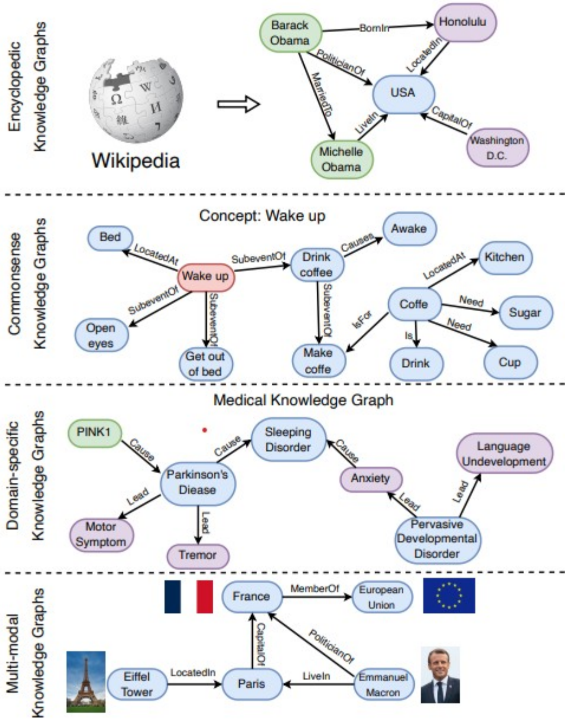
"An Ontology is a description (like a formal specification of a program) of the concepts and relationships that can formally exist for an agent or a community of agents."*



The **MAPE-K** (**Monitor-Analyze-Plan-Execute over a shared Knowledge**) reference control model for autonomic and self-adaptive systems

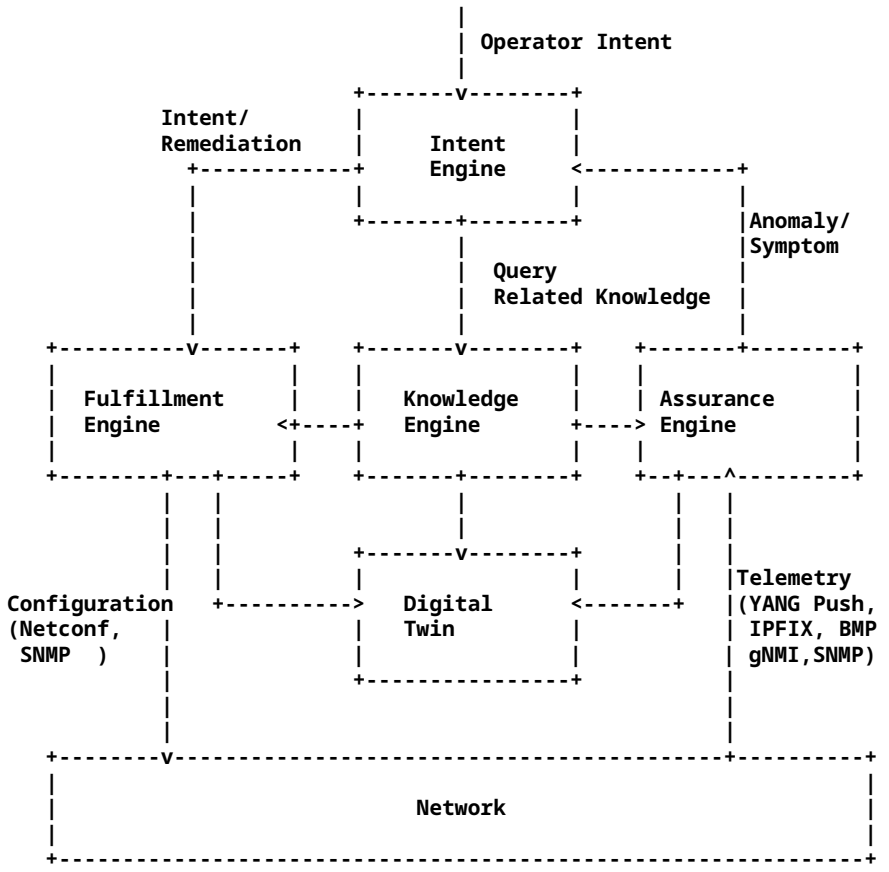


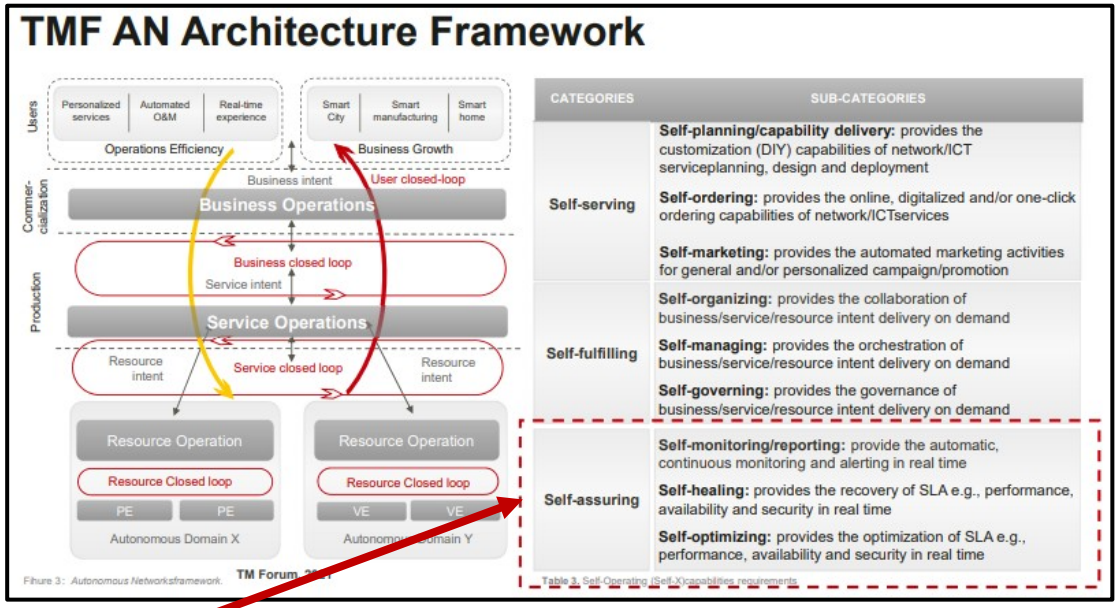
- **Simple structure:** most ontologies describe individuals (**instances**), classes (**concepts**), attributes and relations that can usually be represented at the instance level as (knowledge) graph.
- **Machine readable and machine interpretable:** Information represented in a particular formal ontology can be more easily accessible to automated information processing allowing a machine to connect and understand different entities and their relationships.
- **Is an Ontology a common data model ?:** Yes and No
  - an Ontology and semantic web allow the markup/labeling of existing data but is independent of the data itself, enabling it to work very well with heterogeneous data and for the model to change as your environment and understanding of the world changes without effecting the underlying data itself.
  - Ontologies can define axioms about the domain and from those axioms and the facts already collected can infer new facts.
- **Ontologies are designed to be constantly evolving:** ontologies are not fixed and are designed to continuously change and grow. Different Ontologies can



*Toward principles for the design of ontologies used for knowledge sharing
https://www.sciencedirect.com/science/article/pii/S1071581985710816

Releasing Knowledge Is The Key for autonomic/self-x actions





Knowledge between silos is key to finding deep insight into network data.

- **Relying on the data lake to find relationships and semantic is too late.**
- **Relationship and Semantic knowledge already exists but is hidden inside multiple systems and in multiple formats**
- **How this information is made available to Data Scientists and machine processes is the key to unlock the full potential of the Data Lake and to solve AN**

Challenges For Implementation

- A lot of the opensource tooling originates in academia.
- No “MySQL/Postgres” for RDF data.
  - Virtuoso Opensource is maybe the closest, doesn’t support clustering or high availability https://vos.openlinksw.com/owiki/wiki/VOS
  - We used RDF4j for local development
- Reasoners can result in exponential complex queries (if you are not careful)
- More innovation going into LPG
  - RDF Ecosystem seems small and driven from academia.
  - Neo4j pushing the standardisation of LPG approach (Cypher/GQL)
    - Supports ingestion and serialization to Triples using neosemantic
  - LPG does not have a schema definition language (yet – but no they need one).

Example: Reasoners

Reasoner	OWL 1	OWL 2 DL	OWL 2 EL	OWL 2 RL	Active?	Best For
HermiT		Full	Partial		Oct 2017	Logical completeness, Protégé
Pellet		Partial		Partial	Jan 2017	Legacy projects, SWRL
FaCT++		Partial			2015	TBox reasoning
ELK			Full		May 24	Large bio-ontologies
Konclude		Fast (no datatypes)			Jun 21	Classification performance
RDFox				Full	Active (closed source)	OWL 2 RL, fast, rule-based apps

https://www.w3.org/2001/sw/wiki/OWL/Implementations

Opensource implementations originate in academia and in most cases are no longer actively supported. There are a choice of closed source implementations all come with an * and of course you have to pay

LPG World Is Moving Towards The Semantic

OpenSPG

- Alipay started development on Knowledge graphs in 2018.
- They decided that the Semantic Web stack was too complicated to learn (their estimate = 1 year), LPG are much easier to learn.
- Once they began developing their UCs, they found that in fact they needed the features of Semantic Web to fully describe what they wanted to model and also to allow their model to grow
- Given they had already had invested a huge amount of time and effort, instead of redoing everything in the semantic web they instead created a framework on top of LPG that replicated the Semantic Web Stack, supporting class/property hierarchies, different forms of transitive relationships and a reasoner to interpret the model.
- They since built a LLM layer for Knowledge graph creation (using NER and relationship discovery) called KAG, and an API for LLM interaction for query (using vector similarity and/or query generation.
- A Python SDK layer for interaction with the Knowledge Graph,
- A controller that allows the planning and creation of SPARK pipelines for long running queries.
- Opensource version runs on top of Neo4j but can be adapted to any LPG.

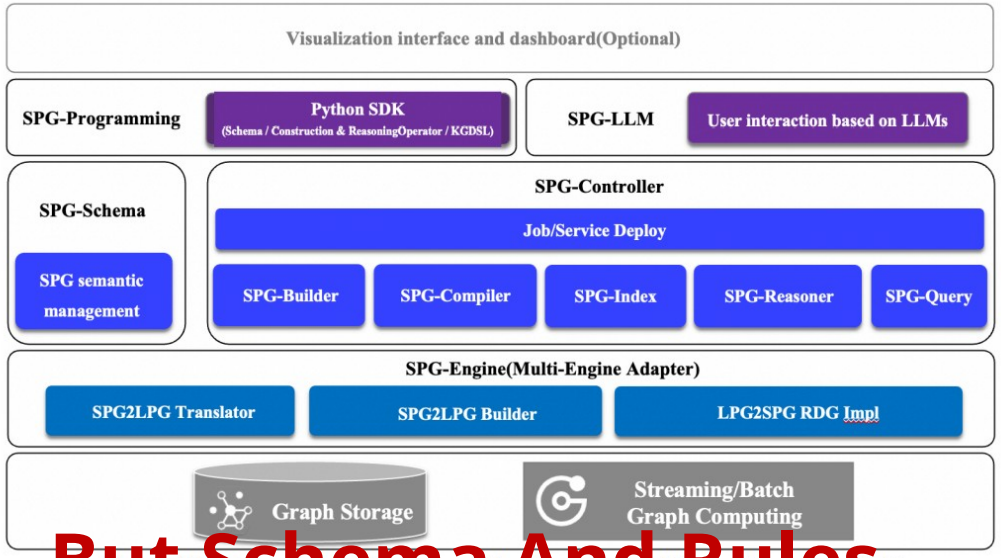
Code: https://github.com/OpenSPG/openspg

Documentation: https://openspg.yuque.com/ndx6g9/docs_en

ANT GROUP

OpenKG

放促进互联，链接创造价

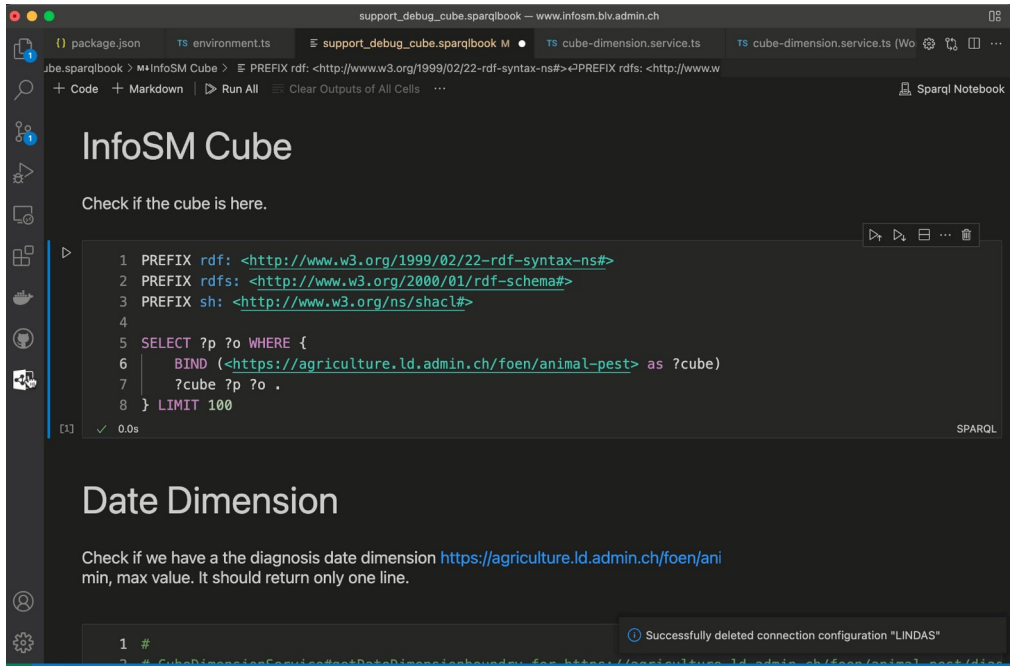


But Schema And Rules Are Still Not Standards Based…

But things in the Ecosystem we (I) did found useful …

SPARQL Book …

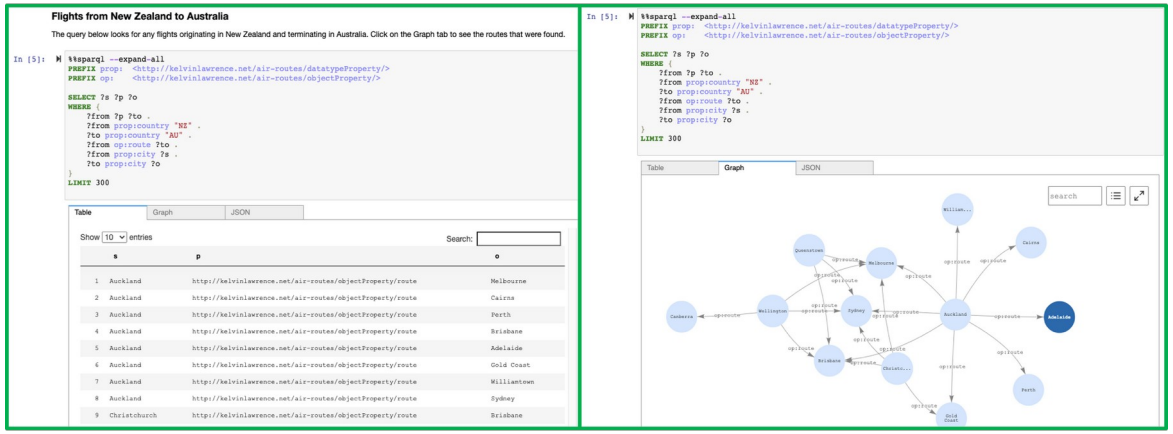
- https://marketplace.visualstudio.com/items?itemName=Zazuko.sparql-notebook



graph-notebook

https://github.com/aws/graph-notebook

Supports cypher, gremlin and SPARQL



What Can The IETF Do ?

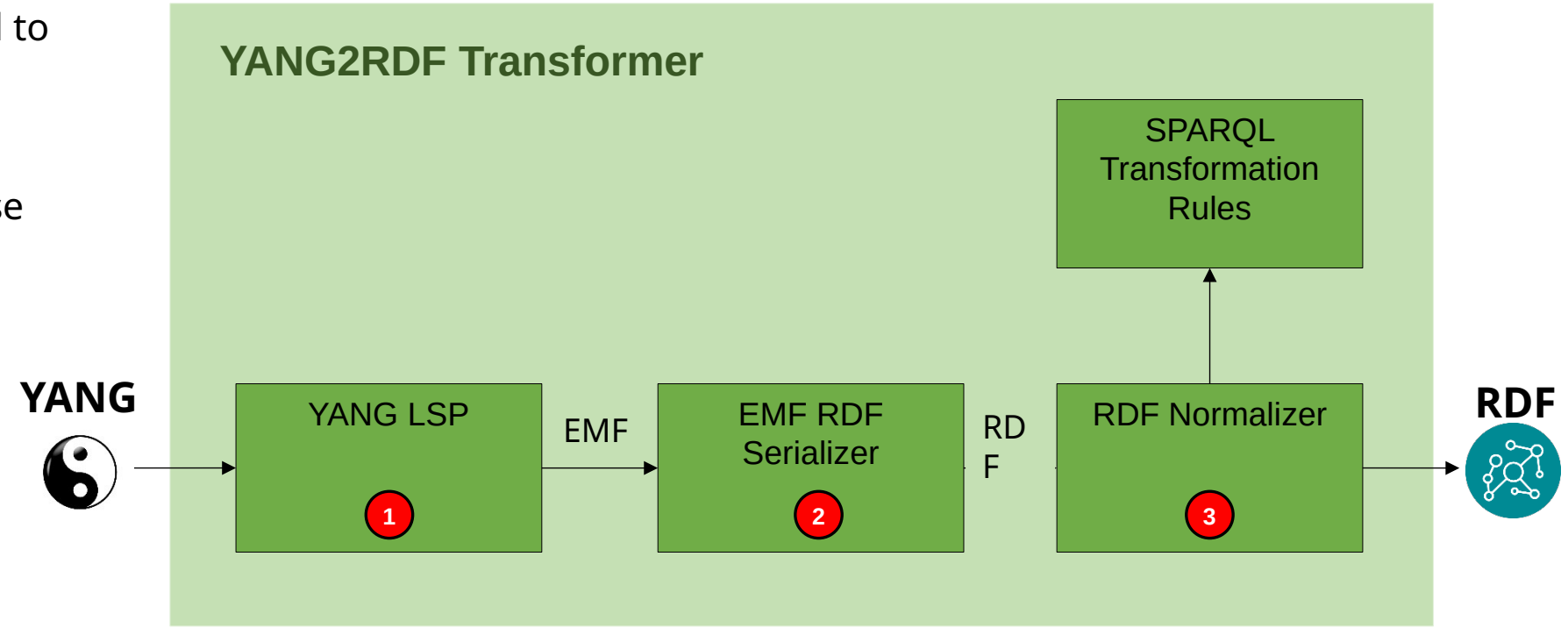
- Formalize a way to share knowledge
  - RFC doesn’t seem a good fit
- Standardise on a implementation independent format for that Knowledge
  - OWL is feature rich but complex and implementations are not homogenous
  - RDFS is less features but is guaranteed to be supported.
- Define a way to capture existing Knowledge (YANG to RDF mapping)
  - Agree a way to define new Knowledge within the IETF ecosystem
- Improve the Draft and evangelise the approach among operators.

Hackaton Results: YANG2RDF Transformer

- YANG model is a hierarchical model in plain text
- YANG model is transformed to RDF using YANG2RDF Transformer
- Anything in RDF can be connected to something else

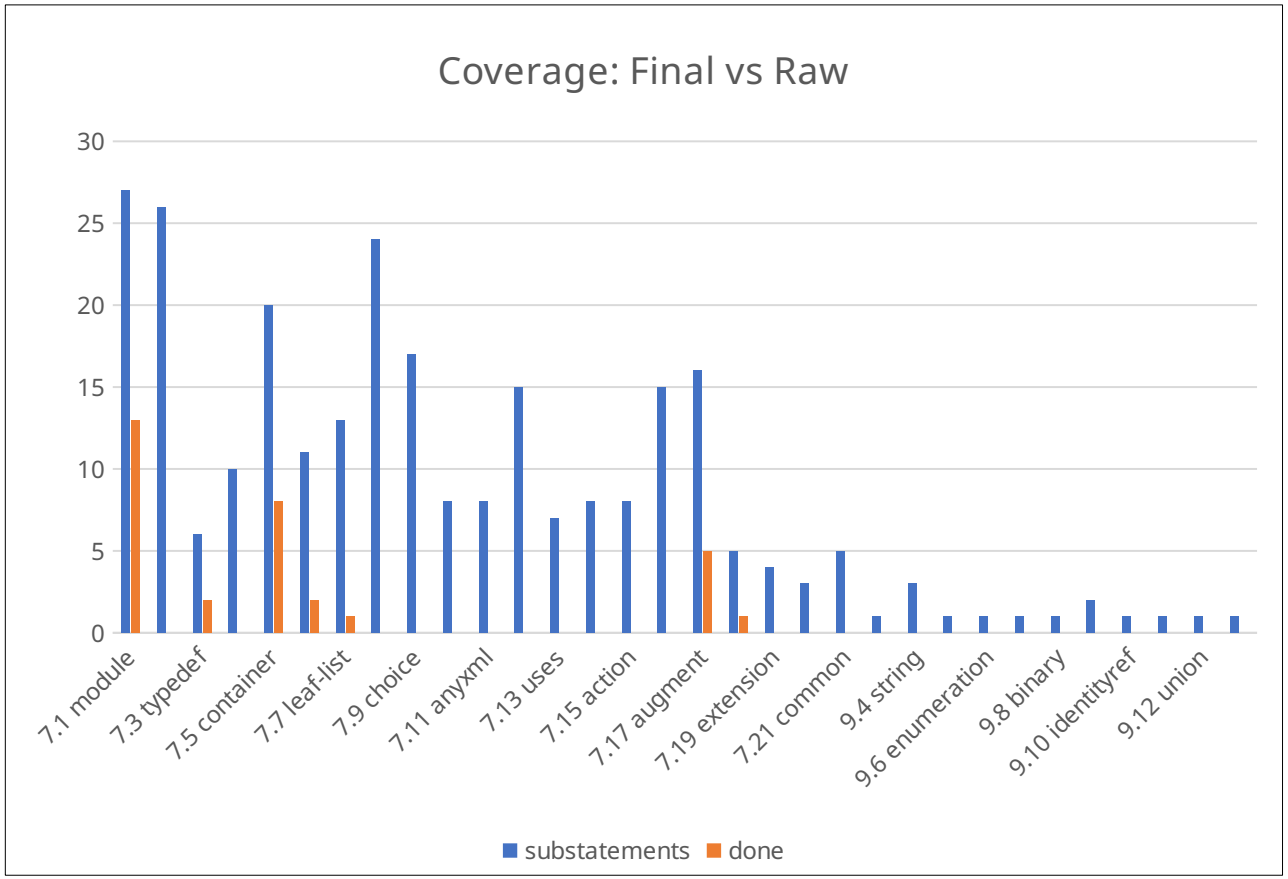
How it works:

1. YANG model is parsed to EMF model
2. EMF model is transformed to RDF
3. RDF is normalized for optimal representation using SPARQL Transformation Rules



IETF 121 Dublin

Hackaton Results: Summary



Full Grammar Support !

- ALL Yang Statements And Substatements translated

Main structural features of YANG Supported

- Reconciliation of Augment, Grouping, Extension, Deviation

Translation Of Autogenerated IRIs to Deterministic IRIs based on XPATH

- Partial support for Module, Container, Leaf, LeafList, Type and Identity

Checkout the code !!! https://github.com/Huawei-IOAM/yang2rdf

No Plan for **IETF 123 Madrid Hackaton**
But ….

We have been working with Eurecom on a YANG 2 RDF Definition and mapping and we hope to publish it in time for the Madrid IETF

Summary

Classes

Module | SubModule | Module Revision | Type Definition | List | Enumeration | Bit | Type | Leaf List | Leaf | Container | Identity | Notification | RPC | Input | Output | AnyData | AnyXML | Choice | Case | Deviation | Deviate | Feature | Extension | Action

Properties

has Revision | has Identity | has RPC | has Feature | has Extension | has Notification | has AnyData | has AnyXML | has Choice | has Deviation | has Container | has Type | has Type Definition | has Enumeration | has Bit | has List | has Leaf | has Leaf List | has Case | has Deviate | has Action | has Input | has Output | Belongs To | Description | Date | Reference | YANG Version | Namespace | Prefix | Organization | Contact | Default | Status | Units | Type | If Feature | Max Elements | Min Elements | Must Condition | Ordered By | Unique | When Condition | Config | Mandatory | Presence | Base | Range | Fraction Digits | Length | Pattern | Require Instance | Position | Value | Must Error App Tag | Must Error Message | Must Description | Must Reference | Argument | Argument Yin Element | Modifier

EURECOM
Sophia Antipolis

HUAWEI