

Integrating Semantic Web Technologies and ASP for Product Configuration

Stefan Bischof, Gottfried Schenner, Simon Steyskal and Richard Taupe
Siemens Corporate Technology





The Knowledge Graph – Things not Strings

<https://www.blog.google/products/search/introducing-knowledge-graph-things-not/>

The image shows a Google search interface for the term "graz". The search bar contains "graz" and the search button is visible. Below the search bar, there are navigation tabs for "All", "Images", "Maps", "News", "Videos", and "More". The search results show "About 56.100.000 results (0,60 seconds)".

A red speech bubble points to the Knowledge Panel with the text: "Knowledge panels use Knowledge Graph API".

Things to do in Graz

 <p>Schlossberg Forested public park with hilltop fort</p>	 <p>Kunsthhaus Graz Amorphous blue modern art museum</p>	 <p>Murinsel River and architecture</p>	 <p>Styrian Armoury Museum of European armor & weaponry</p>
---	---	--	--

[Graz travel guide](#)

Graz - Wikipedia
<https://en.wikipedia.org/wiki/Graz>
Graz is the capital of Styria and the second-largest city in Austria after Vienna. On 1 January 2018, it had a population of 325,021. In 2015, the population of the ...

State: Styria Elevation: 353 m (1,158 ft)
Country: Austria Area codes: +43 316

[Kainbach bei Graz](#) · [Hart bei Graz](#) · [Feldkirchen bei Graz](#) · [Kunsthhaus Graz](#)

Graz
City in Austria

Graz is the capital city of the southern Austrian province of Styria. At its heart is Hauptplatz, the medieval old town's main square. Shops and restaurants line the narrow surrounding streets, which blend Renaissance and baroque architecture. A funicular leads up Schlossberg, the town hill, to the Uhrturm, a centuries-old clock tower. Across the River Mur, futuristic Kunsthhaus Graz exhibits contemporary art.

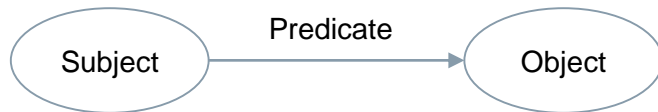
Weather: 22°C, Wind E at 5 km/h, 69 % Humidity
Postal codes: A-801x, A-802x, A-803x, A-804x, A-805x
District: Statutory city
Area codes: +43 316
Population: 269.997 (2014) United Nations

RDF Graph

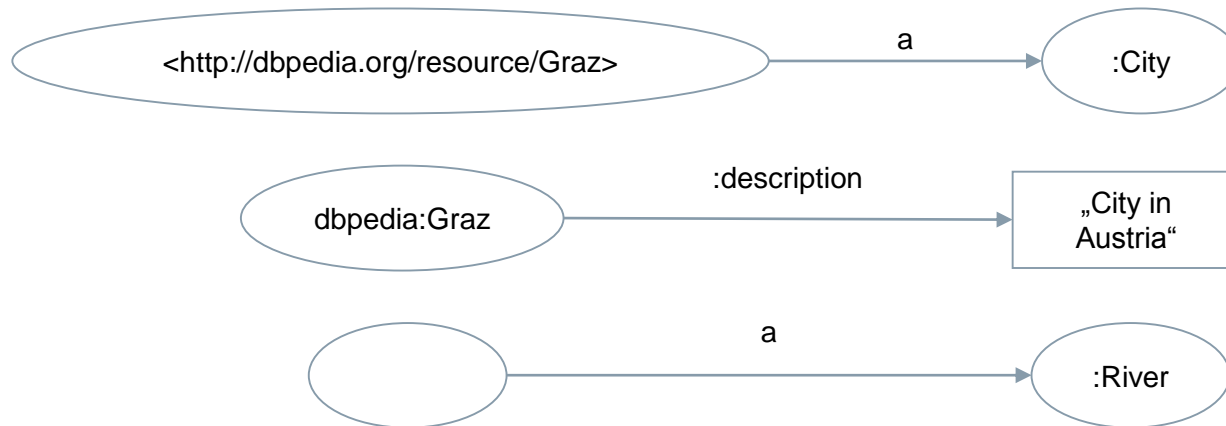
<http://www.w3.org/TR/rdf11-concepts/>

An RDF Graph is a set of RDF Triples

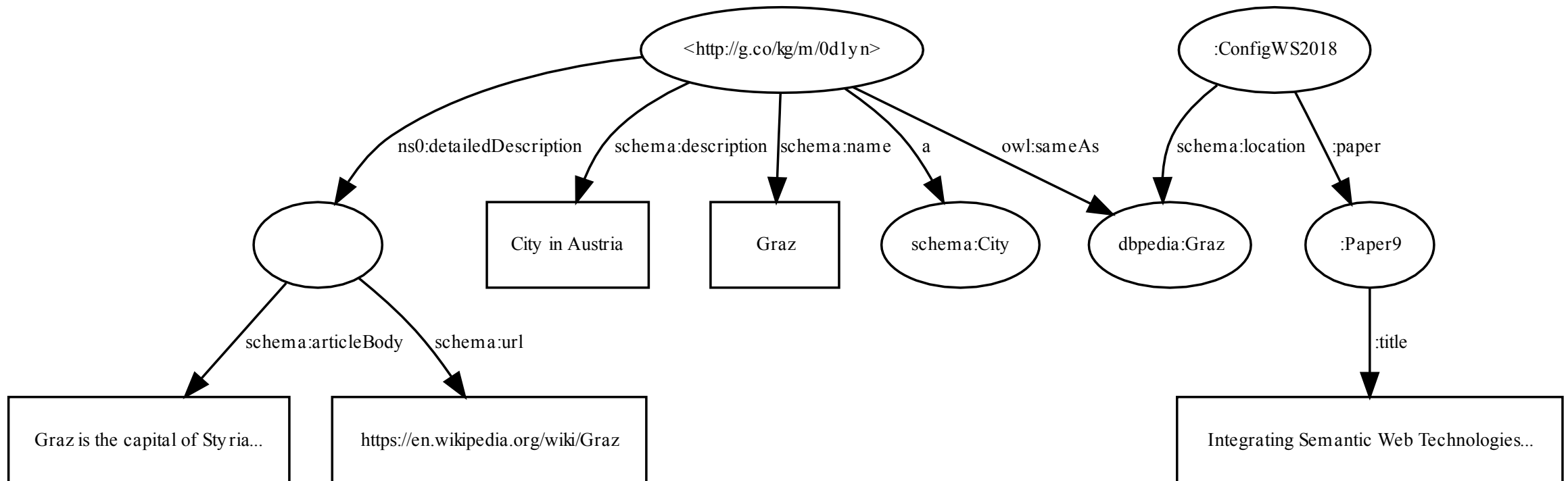
- An RDF triple consists of a subject, a predicate and an object



- An RDF node can be an IRI, a literal or a blank node



RDF Graph Example



Product Configuration with Semantic Web Technologies

HW Configuration Example

In our example domain there may be different types of elements, which are controlled by hardware modules. Each hardware module must be in a frame and a frame must be mounted on a rack.

- There are four disjoint types of elements (ElementA-ElementD).
- There are five disjoint types of modules (ModuleI-ModuleV).
- There are two disjoint types of racks (RackSingle, RackDouble).
- An ElementA/B/C/D requires exactly one/two/three/four ModuleI/II/III/IV respectively.
- A RackSingle must contain exactly four frames, a RackDouble must contain exactly eight frames.
- ...

We use an abstracted toy example. As you see on this picture the real world is more complicated...



Product Configuration KB as RDF Graph

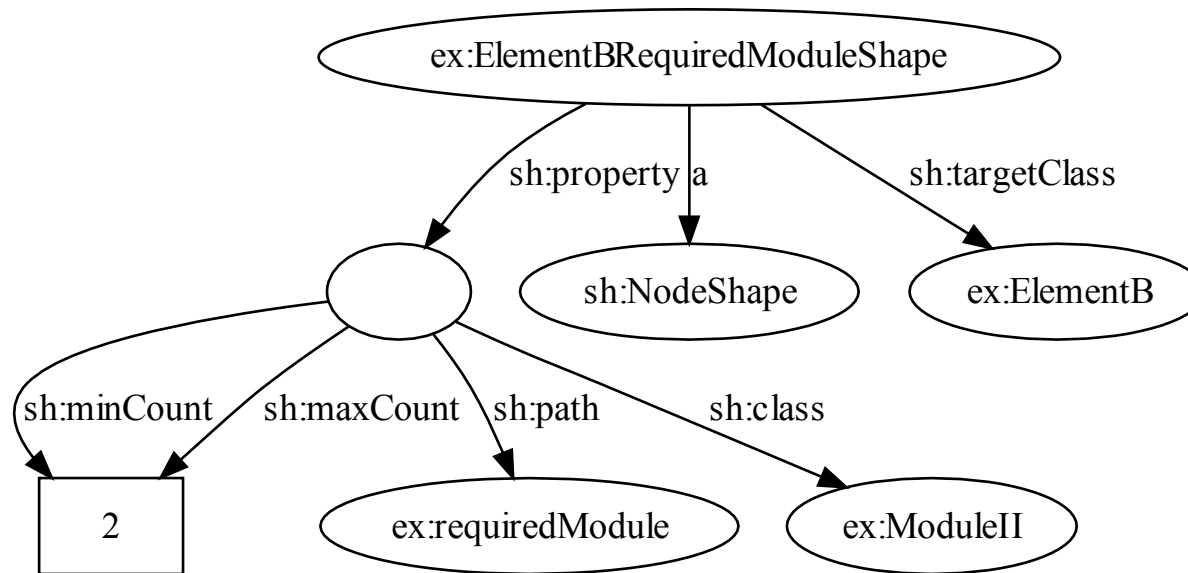


SHACL (Shapes Constraint Language)

<https://www.w3.org/TR/shacl/>

SHACL allows the definition of constraints for RDF Graphs

- The constraints are defined in a SHACL shapes graph



- The shape graph is used to validate a data graph (any RDF graph)
- If you want to try out SHACL: <http://shacl.org/playground/>

Product Configuration with Semantic Web Technologies

Terminology

Definitions:

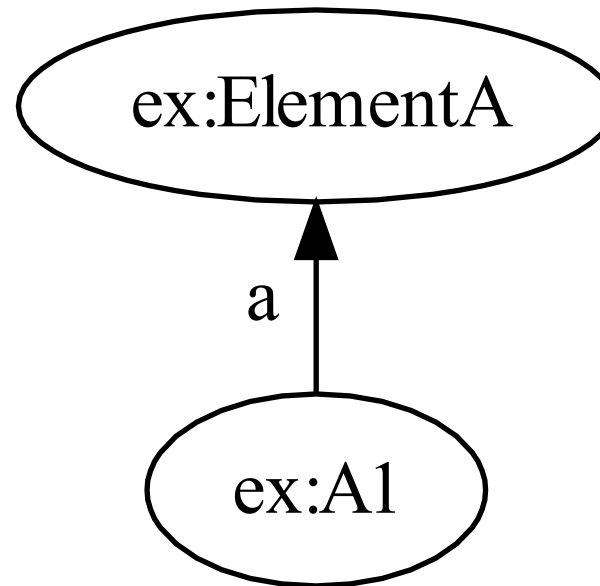
- **Configuration Model:** A SHACL shapes graph and an ontology/RDFS schema defining all the used classes and properties.
- **User Requirements:** consists of additional SHACL constraints and an initial RDF graph.
- **Configuration Task:** The configuration task consists of the configuration model and the user requirements.
- **Configuration:** A configuration (solution) of a configuration task is an RDF graph, which satisfies the SHACL constraints of the configuration model and the user requirements and is an extension of the initial RDF graph.

Terminology according to

*Knowledge-Based Configuration, eds., Alexander Felfernig,
Lothar Hotz, Claire Bagley, and Juha Tiihonen*

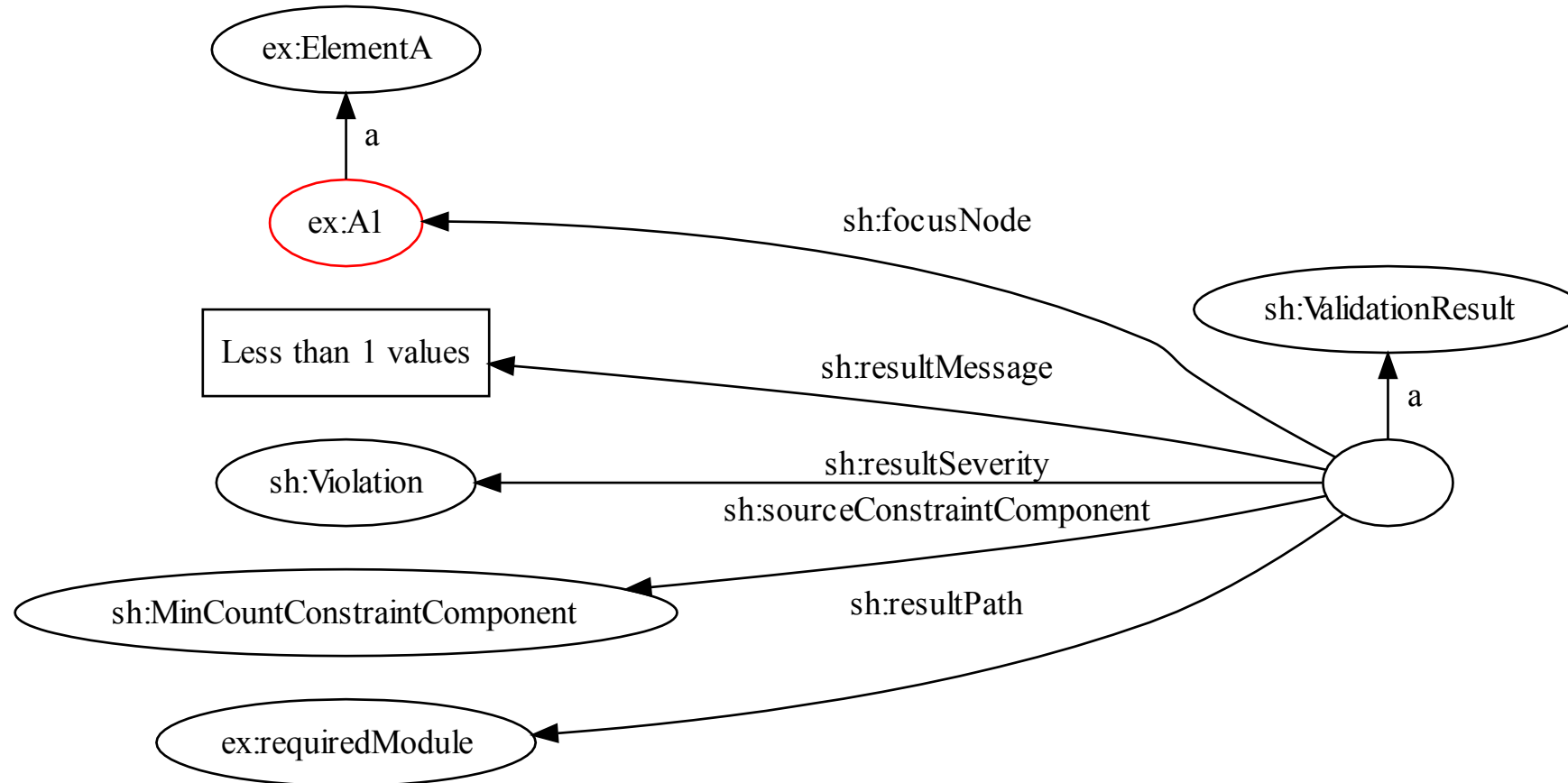
Interactive Product Configuration with SHACL

Example



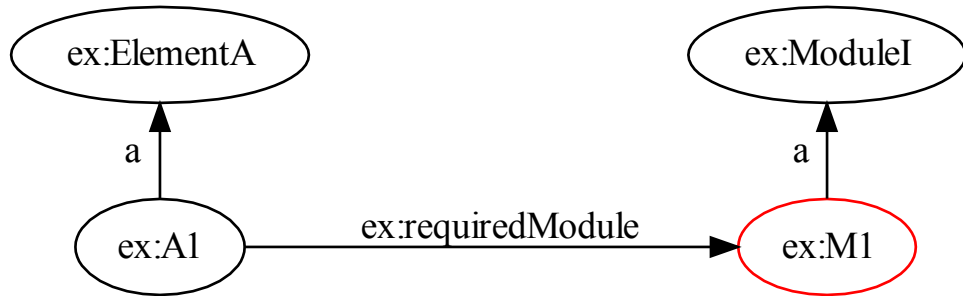
Interactive Product Configuration with SHACL

Example



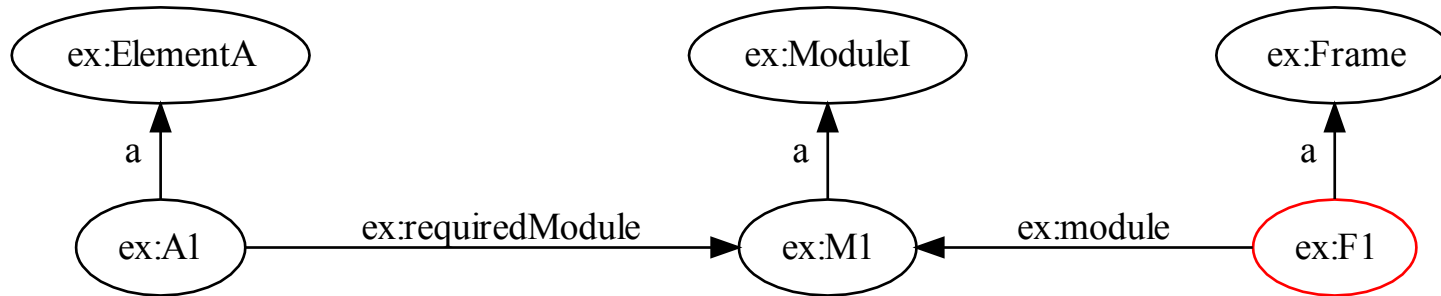
Interactive Product Configuration with SHACL

Example



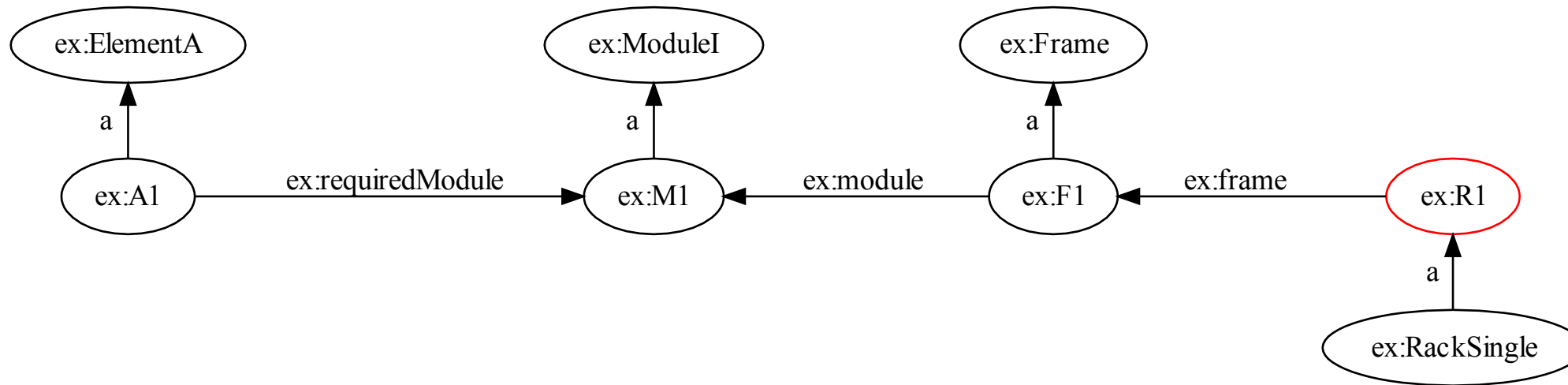
Interactive Product Configuration with SHACL

Example



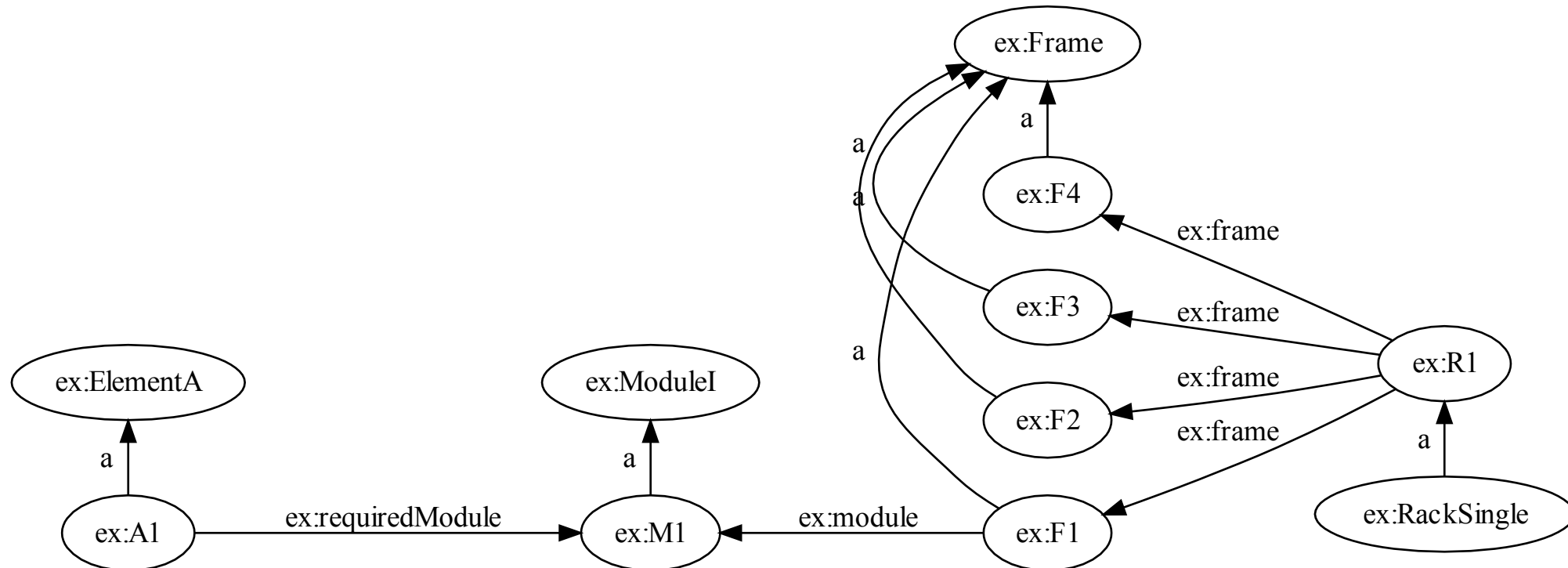
Interactive Product Configuration with SHACL

Example



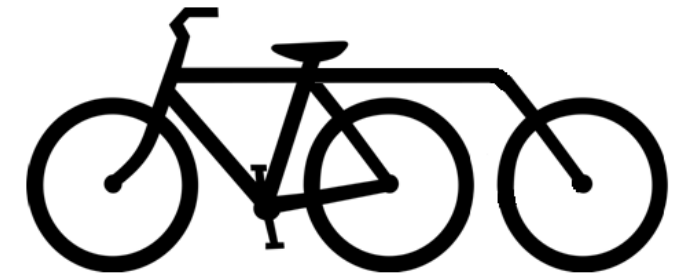
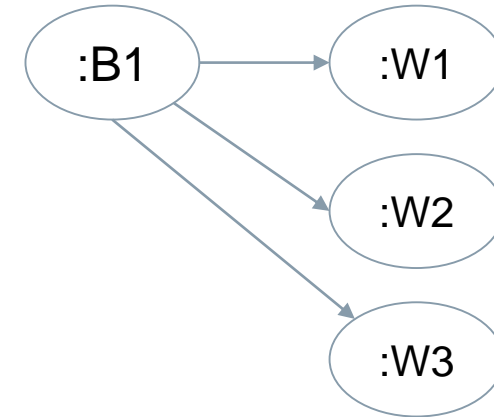
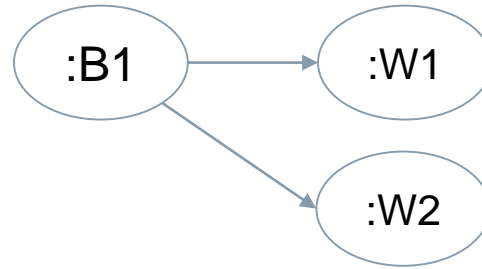
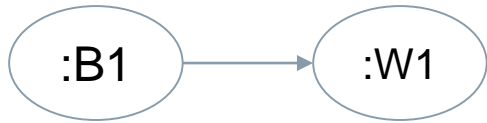
Interactive Product Configuration with SHACL

Example



Why SHACL?

Closed (CWA) vs Open (OWA) world assumption



CWA: 

OWA: 

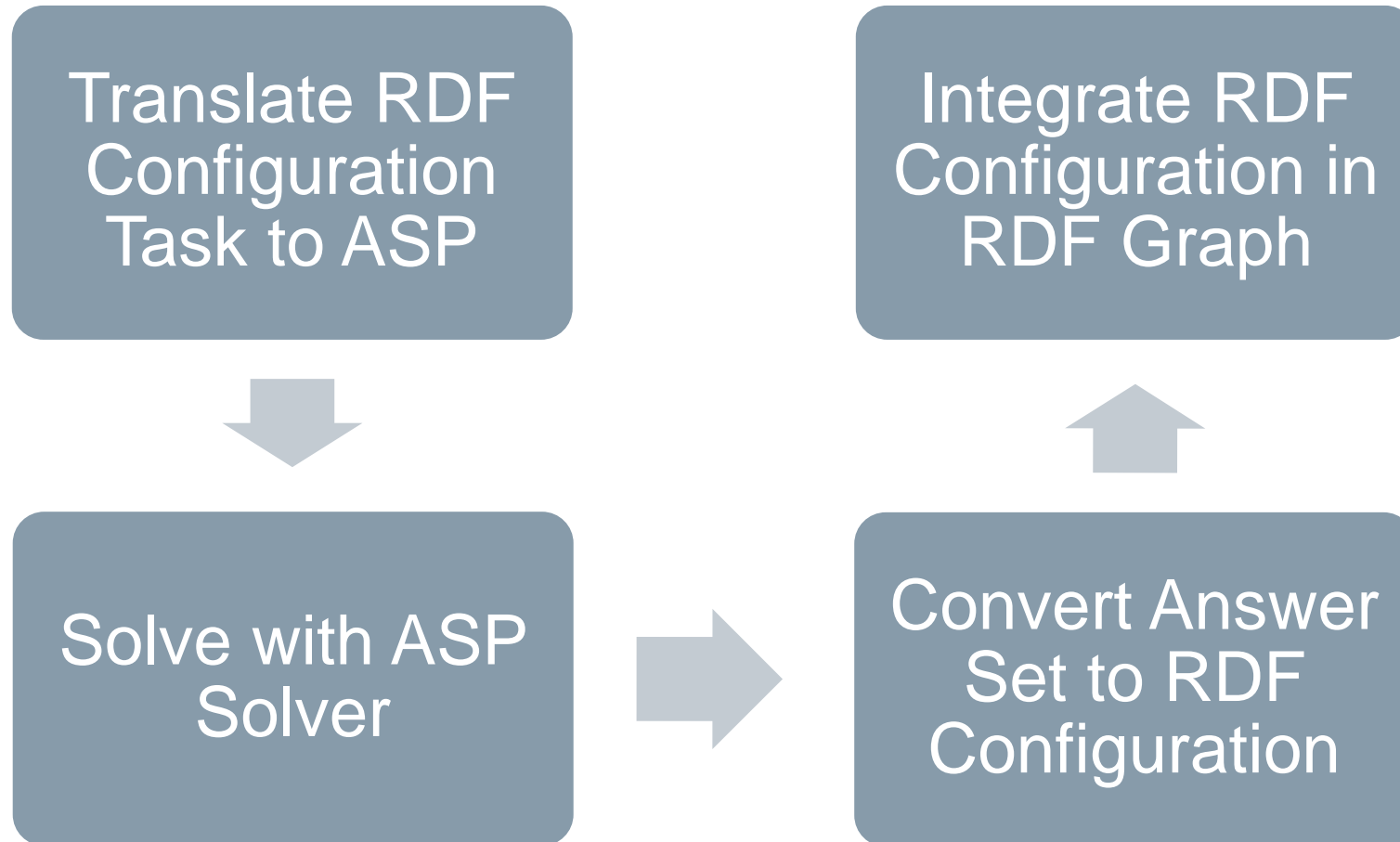
CWA: 

OWA: 

CWA: 

OWA: 

Solving RDF Configuration Task with ASP Overview



Solving with ASP

Main features

- Generic translation of SHACL core constraints into ASP
- Translate configuration task to ASP
 - RDF graph is represented by predicate triple(S,P,O)
 - Potential new individuals are represented as blank nodes
 - Encode specific SHACL constraints manually into ASP
 - If an answer set is found the RDF configuration can easily generated from the triple(S,P,O) atoms.

```
bnode ("_:b1"). bnode ("_:b2"). bnode ("_:b3").
bnode ("_:b4"). bnode ("_:b5"). bnode ("_:b6").
0 { triple (BNODE , "a", "ex:RackSingle") } 1 :-
    bnode ( BNODE ).
0 { triple (BNODE , "a", "ex:RackDouble") } 1 :-
    bnode ( BNODE ).
...
% answer set found
triple("_:b1","a","ex:Rack").
triple("_:b1","a","ex:RackSingle").
triple("_:b2","a","ex:Frame").
triple("_:b3","a","ex:Frame").
triple("_:b4","a","ex:Frame").
triple("_:b5","a","ex:Frame").
triple("_:b1","ex:frame ","ex:b2").
triple("_:b1","ex:frame ","ex:b3").
triple("_:b1","ex:frame ","ex:b4").
triple("_:b1","ex:frame ","ex:b5").
```

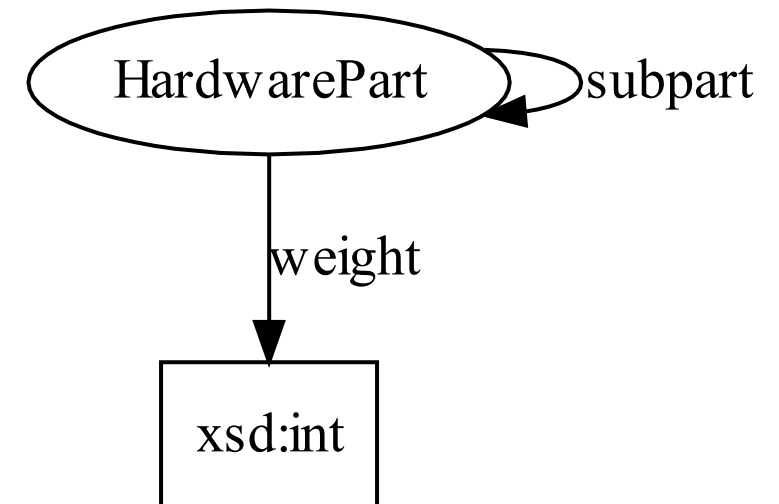
OWL (Web Ontology Language)

<http://www.w3.org/TR/owl2-overview/>

Use OWL to integrate the RDF configuration into the Knowledge Graph

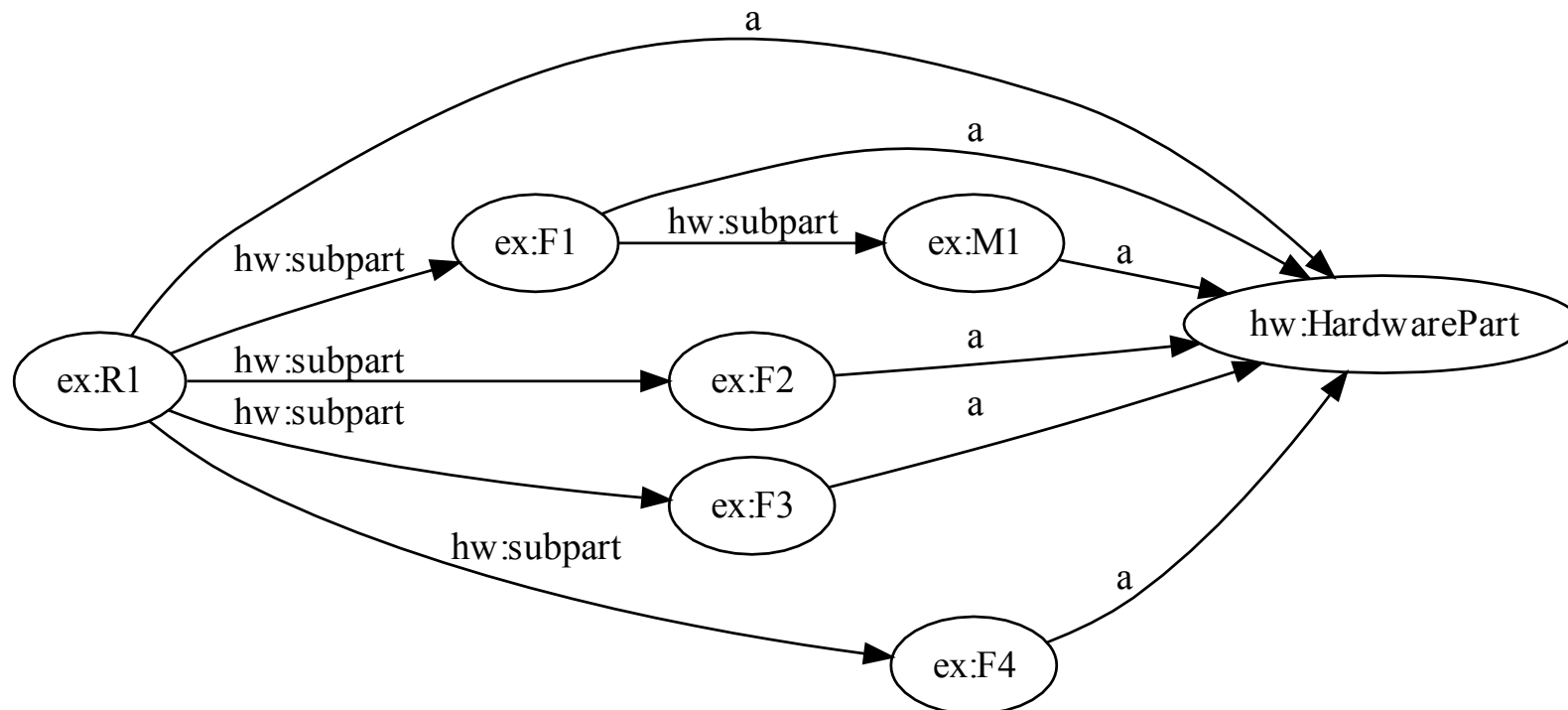
Example reuse of existing ontologies

- Suppose there is an existing generic hardware ontology that allows the computation of the aggregated weight of hardware parts.
- How can we reuse that hardware ontology for our example?
- Solution:
 - Use OWL reasoner to classify our example ontology in terms of the HardwarePart ontology
 - It is sufficient to state which properties constitute the subpart property.



Interactive Product Configuration with SHACL

Example



Some Ideas for Future Work

- Create more complex product configuration models with SHACL
- Automatic translation from SHACL to ASP or other solving paradigms
- Define reusable product configuration ontologies
- Extract product configuration ontologies from documentation with NLP
- ...
- Hopefully work on all these topics together with the product configuration community and share ideas in the Open Source GIT repository

The examples of the paper can be found in the Open Source Siemens GIT-Repository:

<https://github.com/siemens/ProductConfigurationWithSHACL>

Thank you for your attention!

SIEMENS
Ingenuity for life

Gottfried Schenner
Research Scientist
Siemens Corporate Technology

E-mail:
gottfried.schenner@siemens.com

Internet
[siemens.com/corporate-technology](https://www.siemens.com/corporate-technology)