

Common Cognitive Radio language, a CREW-GENI collaboration

Carolina Fortuna¹, Milorad Tomic², Mikolaj Chwalisz³, Peter de Valck⁴, Ivan Seskar⁵

¹ Jozef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia

² University of Nis, Serbia

³ Technical University Berlin, Germany

⁴ iMinds and University of Gent, Belgium

⁵ Rutgers University, USA.

Outline

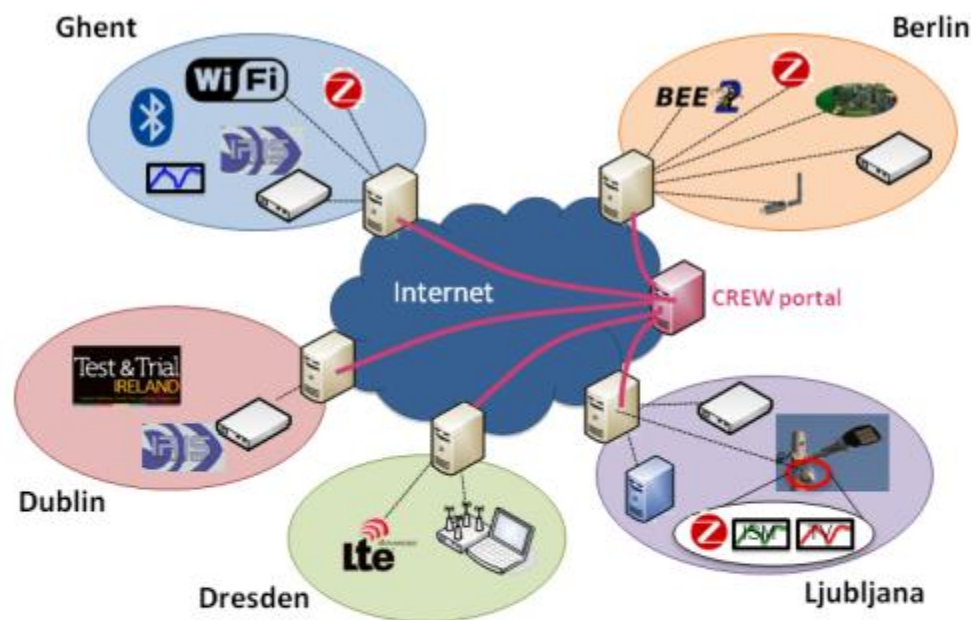
- Introduction: CREW CDF and TassOR
- About ontologies and CR use cases
- The Spectrum Sensing Ontology
- Modelling and querying cognitive radio devices: selected examples
- Summary

The CREW federation's common data format

- 5 testbeds
- Need for harmonizing experiment description
- Need for harmonizing the results from the experiment



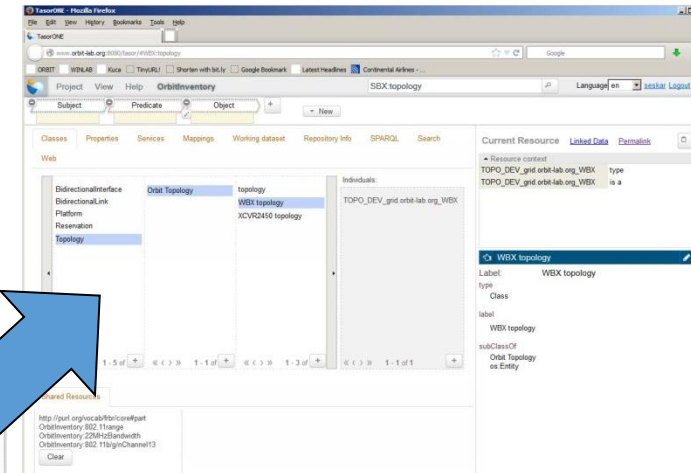
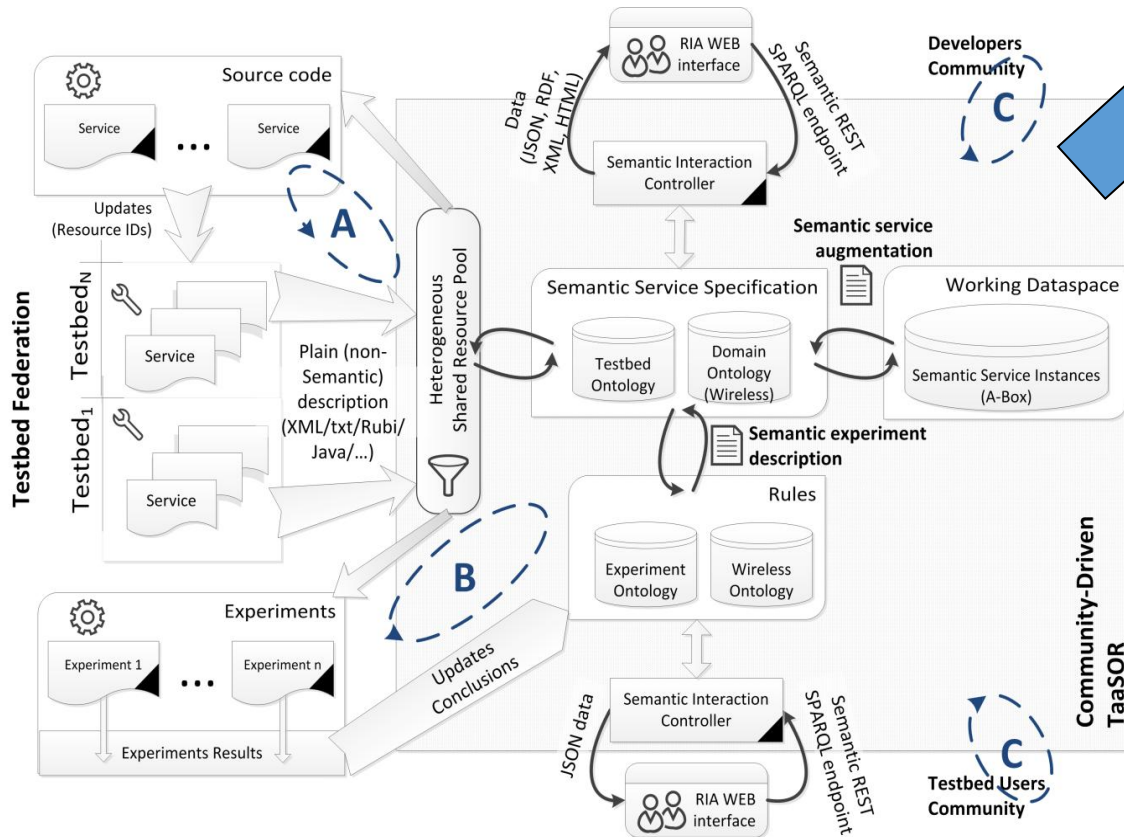
common data format that includes a vocabulary based on the IEEE P 1900.1 specifications



WiFi	IEEE 802.11	IRIS GPP-based software radio platform	IMEC Sensing Agent
IEEE 802.15.1		Comreg spectrum licenses	UHF/VHF TV sensing
IEEE 802.15.4		BEE2 FPGA platform	ISM bands sensing
LTE-advanced		USRP software radio	TCS Multi-antenna LTE detection
EyesIFX nodes		VESNA platform on light pole	WiSpy Spectrum analyzer
CR database			Interconnection of portals
			Interconn. between testbed elements

Testbed as a Service with Ontology Repository (TaaSOR)

Community focused WEB based tool



- Grounding by team consensus on shared ontologies
- Knowledge reuse by including complementing networking ont. (NOVI, NDL, NDL-OWL)
- Generated ontologies instantly available online and published as Linked Data as well as SPARQL endpoint

CREW-GENI efforts

- Starting from
 - existing standard vocabularies provided by ontologies (NOVI, NDL, NDL-OWL)
 - the existing Testbed as a Service with Ontology Repository (TaaSOR)
 - and the CREW common data format
 - create a Spectrum Sensing Ontology capable of describing any device and capability from any cognitive radio testbed
- The final goal is to enable testbed-as-a-service functionality

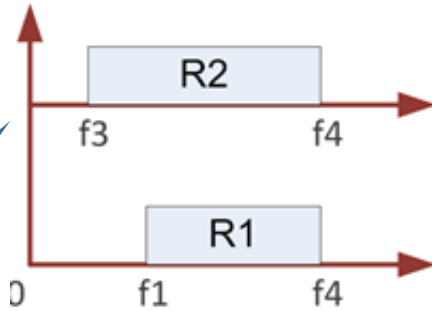
What is an ontology and why semantic web?

- “an ontology formally represents knowledge as a set of concepts within a domain, using a shared vocabulary to denote the types, properties and interrelationships of those concepts”
- In a way, an ontology is a taxonomy that also has relationships between the concepts
- Ontologies and semantic web technologies enable machine interpretable representations of data thus increasing interoperability
- In semantic web, everything can be described as a triple:

Subject (Resource)	Predicate (Property)	Object (Statement)
sses:RadioDevice	rdf:hasIndividual	sses:SpectrumAnalyzerTest
sses:SpectrumAnalyzerTest	sses:supports	sses:SASettings1

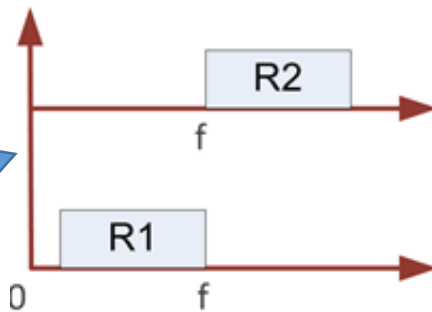
Example use case for CR: model relationships between frequency intervals

Situations we would like to be able to represent



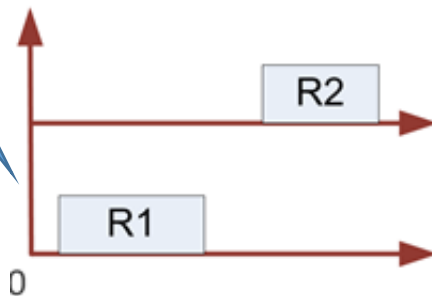
R1 is a `wo:ProperRange`
R2 is a `wo:ProperRange`
R1 `wo:hasStartFrequency` f1
R2 `wo:hasStartFrequency` f3

R1 `wo:hasEndFrequency` f4
R2 `wo:hasEndFrequency` f4
f3 `wo:lower` f1
R1 `wo:rangeEndMatch` R2



R1 is a `wo:ProperRange`
R2 is a `wo:ProperRange`
R1 `wo:endFrequency` f
R2 `wo:startFrequency` f

R1 `wo:rangeAdjacent` R2



R1 is a `wo:ProperRange`
R2 is a `wo:ProperRange`
R1 `wo:lower` R2

R1 `wo:rangeLower` R2

Possible semantic rules for expressing the relationships about frequency ranges

```

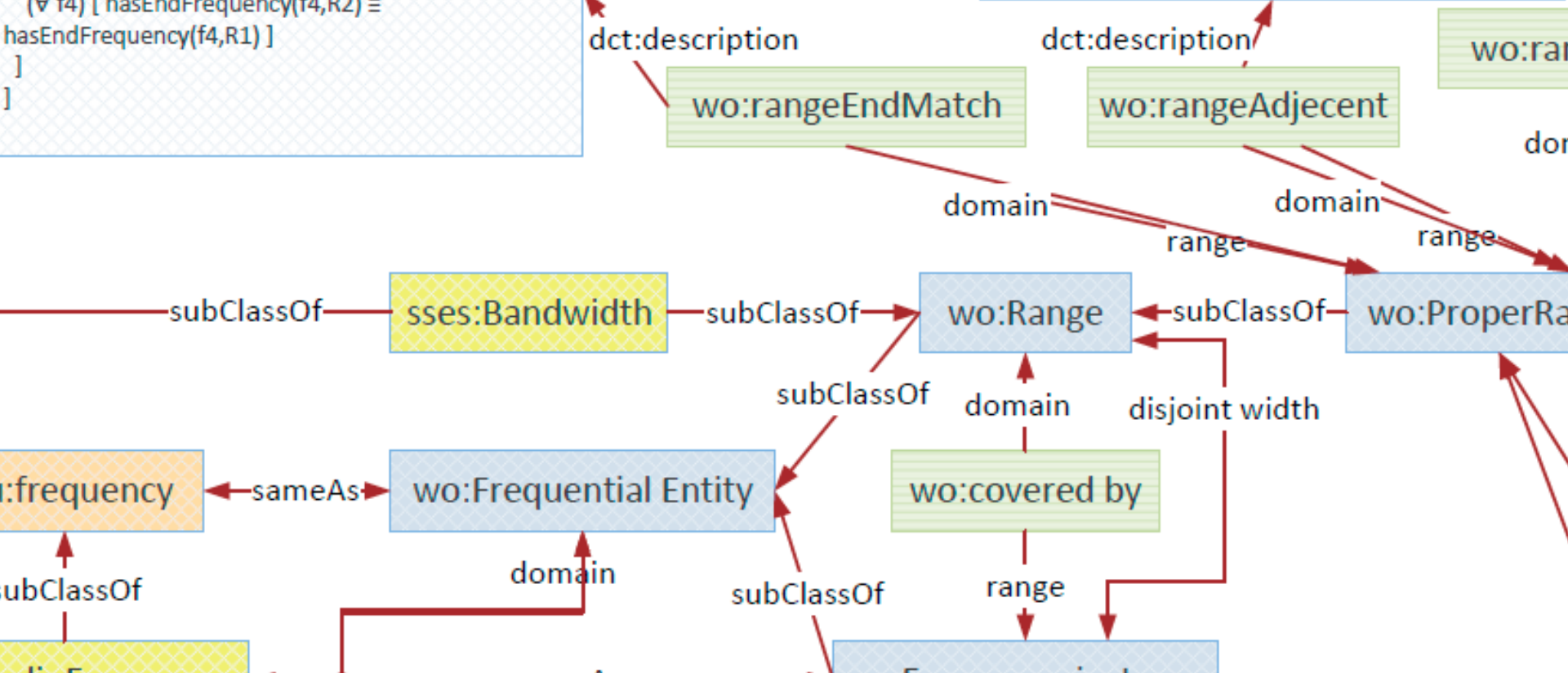
(∀ R1,R2) [
  rangeEndMatch(R1,R2) ≡ [
    ProperRange(R1) ∧ ProperRange(R2) ∧
    (∃ f1) [ hasStartFrequency(f1 ,R1) ∧
      (∀ f3) [hasStartFrequency(f3,R2) ⇒lower(f3,f1) ] ∧
    (∀ f4) [ hasEndFrequency(f4,R2) ≡
      hasEndFrequency(f4,R1) ]
  ]
]

```

```

(∀ R1,R2) [
  rangeAdjacent(R1 ,R2) ≡ [
    ProperRange(R1) ∧ ProperRange(R2) ∧
    (∃ f)[endFrequency(f,R1) ∧ startFrequency(f,R2)]
  ]
]

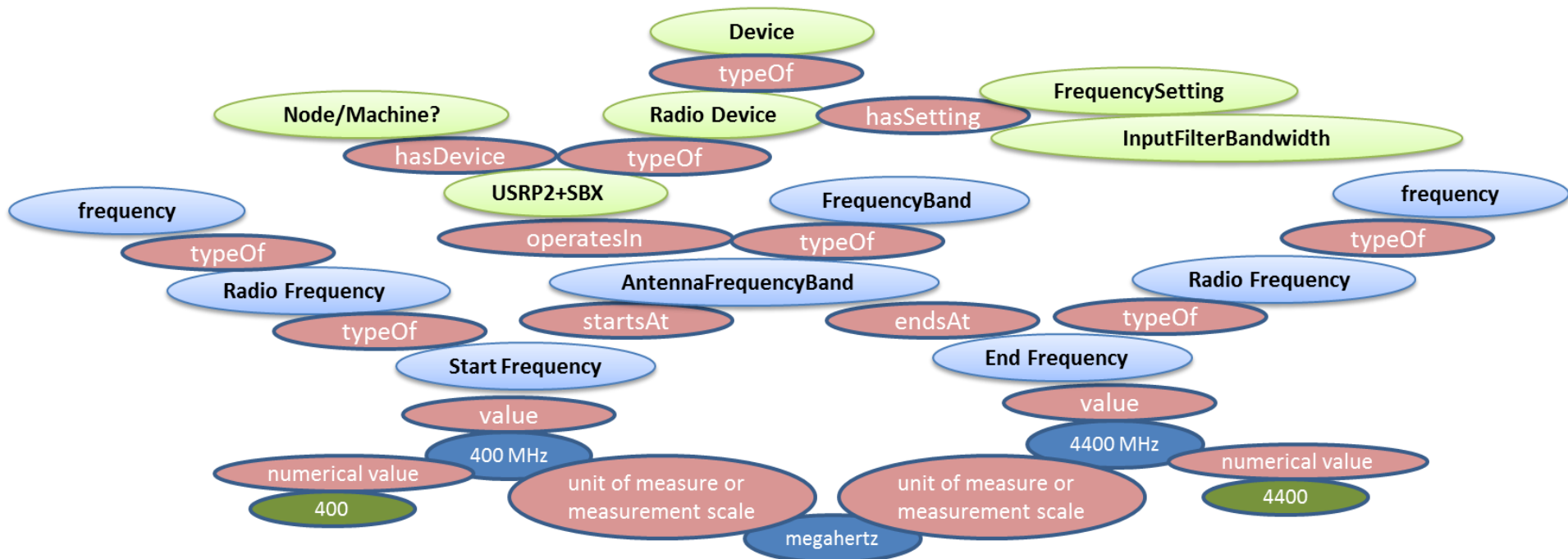
```



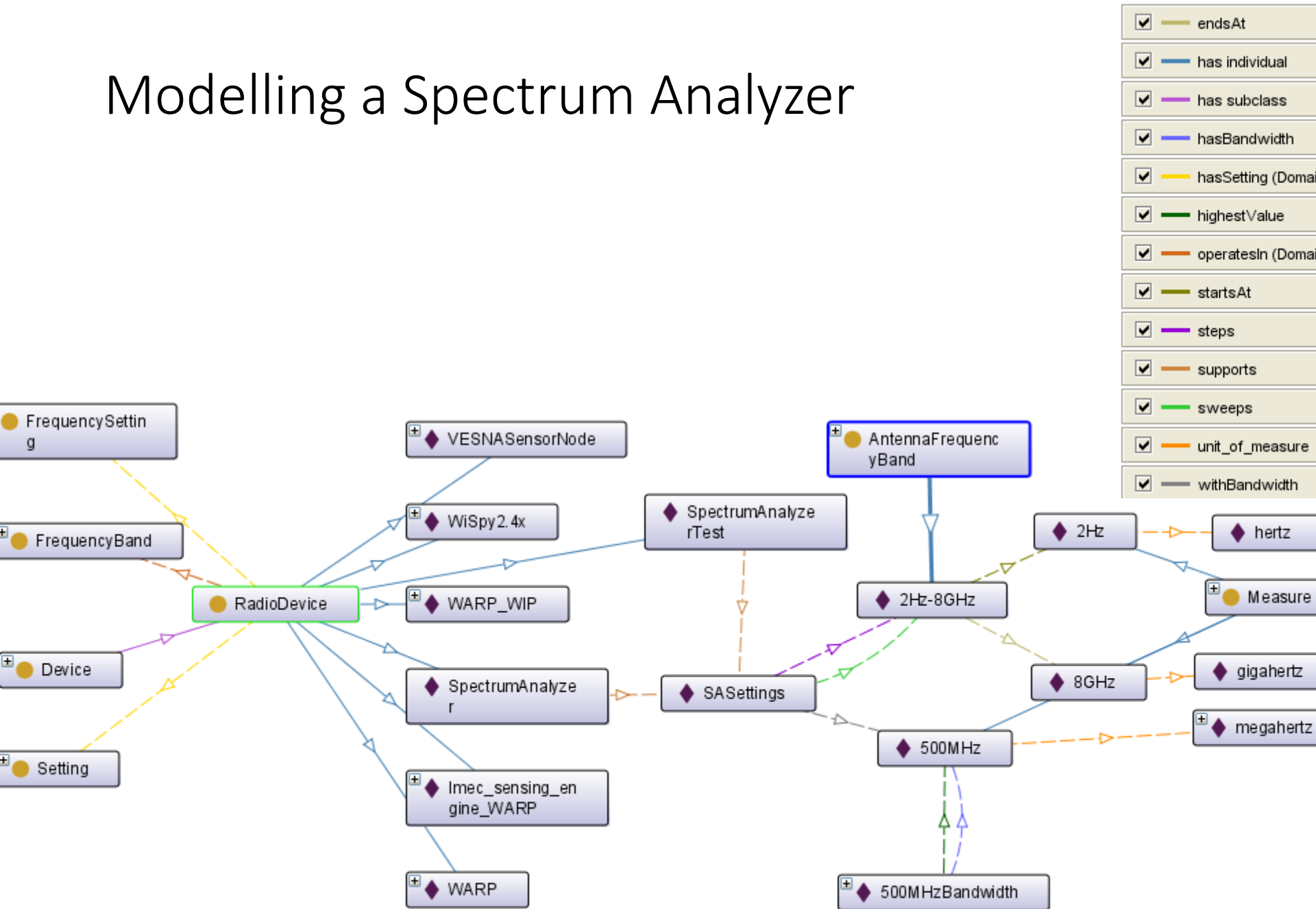
The Spectrum Sensing Ontology (1/2)

- The ontology has three orthogonal parts that allow the description of:
 - spectrum related theoretical aspects,
 - device spectrum sensing capabilities and
 - ranges of values for each
- Basic device capability description: base band and RF capabilities described
- Description of the processing for base
- Current version:
<http://sensorlab.ijs.si/2013/v0/SpectrumSensingExperimentSpecification.owl>

The Spectrum Sensing Ontology (2/2)

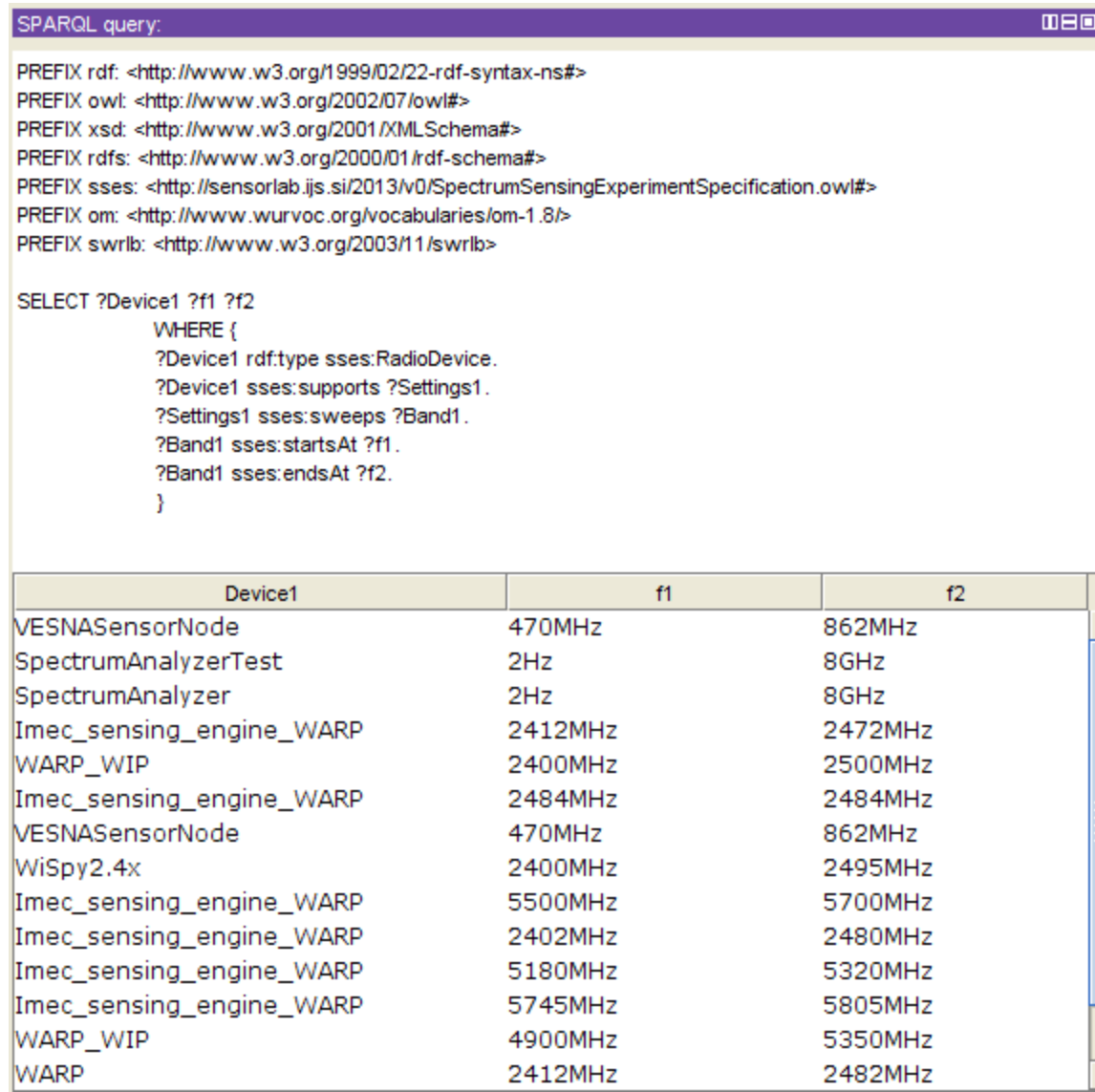


Modelling a Spectrum Analyzer



Example query: Devices and their sweeping bands

- A set of devices have been added as individuals of the concepts in the ontology
- Their properties were also modelled
- A SPARQL query retrieves all the devices and the frequency ranges they are able to sweep



The screenshot shows a SPARQL query interface with a purple header. The query is as follows:

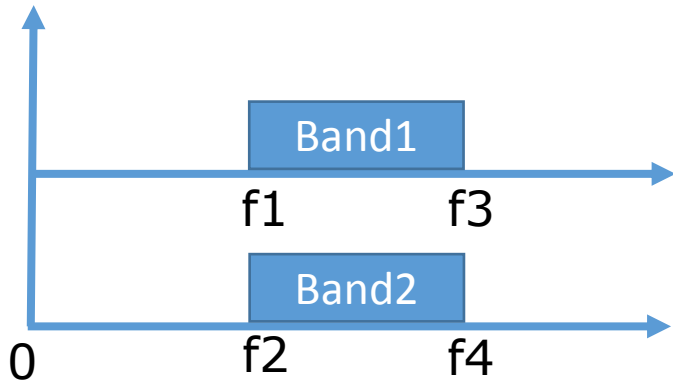
```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sses: <http://sensorlab.ijs.si/2013/v0/SpectrumSensingExperimentSpecification.owl#>
PREFIX om: <http://www.wurvoc.org/vocabularies/om-1.8/>
PREFIX swrlb: <http://www.w3.org/2003/11/swrlb>

SELECT ?Device1 ?f1 ?f2
WHERE {
  ?Device1 rdf:type sses:RadioDevice.
  ?Device1 sses:supports ?Settings1.
  ?Settings1 sses:sweeps ?Band1.
  ?Band1 sses:startsAt ?f1.
  ?Band1 sses:endsAt ?f2.
}
```

The results are displayed in a table with three columns: Device1, f1, and f2.

Device1	f1	f2
VESNASensorNode	470MHz	862MHz
SpectrumAnalyzerTest	2Hz	8GHz
SpectrumAnalyzer	2Hz	8GHz
Imec_sensing_engine_WARP	2412MHz	2472MHz
WARP_WIP	2400MHz	2500MHz
Imec_sensing_engine_WARP	2484MHz	2484MHz
VESNASensorNode	470MHz	862MHz
WiSpy2.4x	2400MHz	2495MHz
Imec_sensing_engine_WARP	5500MHz	5700MHz
Imec_sensing_engine_WARP	2402MHz	2480MHz
Imec_sensing_engine_WARP	5180MHz	5320MHz
Imec_sensing_engine_WARP	5745MHz	5805MHz
WARP_WIP	4900MHz	5350MHz
WARP	2412MHz	2482MHz

Example query: Devices with same sweeping bands

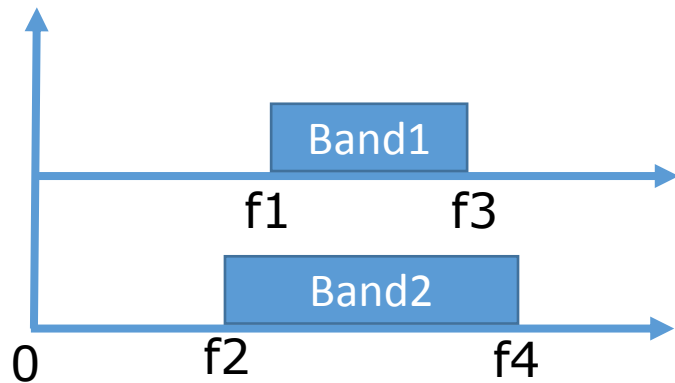


```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sses: <http://sensorlab.ijs.si/2013/v0/SpectrumSensingExperimentSpecification.owl#>
PREFIX om: <http://www.wurvoc.org/vocabularies/om-1.8/>
PREFIX swrlb: <http://www.w3.org/2003/11/swrlb>

SELECT ?Device1 ?Device2 ?Band1 ?Band2
WHERE {
  ?Device1 rdf:type sses:RadioDevice.
  ?Device2 rdf:type sses:RadioDevice.
  ?Device1 sses:supports ?Settings1.
  ?Device2 sses:supports ?Settings2.
  ?Settings1 sses:sweeps ?Band1.
  ?Settings2 sses:sweeps ?Band2.
  ?Band1 sses:startsAt ?f1.
  ?Band2 sses:startsAt ?f2.
  ?Band1 sses:endsAt ?f3.
  ?Band2 sses:endsAt ?f4.
  ?f1 sses:numericalValue ?x.
  ?f2 sses:numericalValue ?y.
  ?f3 sses:numericalValue ?z.
  ?f4 sses:numericalValue ?t.
  FILTER(xsd:integer(?x) = xsd:integer(?y)).
  FILTER(xsd:integer(?z) = xsd:integer(?t)).
}
```

Device1	Device2	Band1	Band2
VESNASensorNode	VESNASensorNode	470-862MHz	470-862MHz
VESNASensorNode	VESNASensorNode	470-862MHz	470-862MHz
SpectrumAnalyzerTest	SpectrumAnalyzerTest	2Hz-8GHz	2Hz-8GHz
SpectrumAnalyzerTest	SpectrumAnalyzer	2Hz-8GHz	2Hz-8GHz
SpectrumAnalyzer	SpectrumAnalyzerTest	2Hz-8GHz	2Hz-8GHz
SpectrumAnalyzer	SpectrumAnalyzer	2Hz-8GHz	2Hz-8GHz
Imec_sensing_engine_WARP	Imec_sensing_engine_WARP	2412-2472MHz	2412-2472MHz
WARP_WIP	WARP_WIP	2400-2500MHz	2400-2500MHz
Imec_sensing_engine_WARP	Imec_sensing_engine_WARP	2484-2484MHz	2484-2484MHz
VESNASensorNode	VESNASensorNode	470-862MHz	470-862MHz
VESNASensorNode	VESNASensorNode	470-862MHz	470-862MHz
WiSpy2.4x	WiSpy2.4x	2400-2495MHz	2400-2495MHz

Example query: Devices with included sweeping bands



Device1	Device2	Band1	Band2
WARP_WIP	Imec_sensing_engine_WARP	2400-2500MHz	2412-2472MHz
WARP_WIP	Imec_sensing_engine_WARP	2400-2500MHz	2484-2484MHz
WARP_WIP	Imec_sensing_engine_WARP	2400-2500MHz	2402-2480MHz
WARP_WIP	WARP	2400-2500MHz	2412-2482MHz
WiSpy2.4x	Imec_sensing_engine_WARP	2400-2495MHz	2412-2472MHz
WiSpy2.4x	Imec_sensing_engine_WARP	2400-2495MHz	2484-2484MHz
WiSpy2.4x	Imec_sensing_engine_WARP	2400-2495MHz	2402-2480MHz
WiSpy2.4x	WARP	2400-2495MHz	2412-2482MHz
Imec_sensing_engine_WARP	Imec_sensing_engine_WARP	2402-2480MHz	2412-2472MHz
WARP_WIP	Imec_sensing_engine_WARP	4900-5350MHz	5180-5320MHz
Imec_sensing_engine_WARP	Imec_sensing_engine_WARP	2400-2485MHz	2412-2472MHz
Imec_sensing_engine_WARP	Imec_sensing_engine_WARP	2400-2485MHz	2484-2484MHz
Imec_sensing_engine_WARP	Imec_sensing_engine_WARP	2400-2485MHz	2402-2480MHz
Imec_sensing_engine_WARP	WARP	2400-2485MHz	2412-2482MHz

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sses: <http://sensorlab.ijs.si/2013/v0/SpectrumSensor#>
PREFIX om: <http://www.wurvoc.org/vocabularies/om-1.8#>
PREFIX swrlb: <http://www.w3.org/2003/11/swrlb#>
```

```
SELECT ?Device1 ?Device2 ?f1 ?f2 ?f3 ?f4
WHERE {
  ?Device1 rdf:type sses:RadioDevice.
  ?Device2 rdf:type sses:RadioDevice.
  ?Device1 sses:supports ?Settings1.
  ?Device2 sses:supports ?Settings2.
  ?Settings1 sses:sweeps ?Band1.
  ?Settings2 sses:sweeps ?Band2.
  ?Band1 sses:startsAt ?f1.
  ?Band2 sses:startsAt ?f3.
  ?Band1 sses:endsAt ?f2.
  ?Band2 sses:endsAt ?f4.
  ?f1 sses:numericalValue ?x.
  ?f2 sses:numericalValue ?y.
  ?f3 sses:numericalValue ?z.
  ?f4 sses:numericalValue ?t.
  FILTER(xsd:integer(?x) < xsd:integer(?z)).
  FILTER(xsd:integer(?t) < xsd:integer(?y)).
}
```

Rules to simplify the SPARQL queries and abstract the underlying complexity

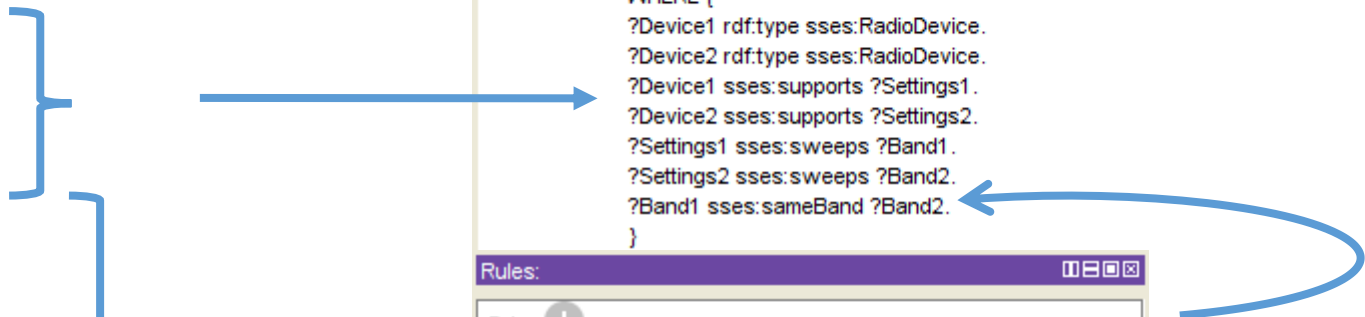
```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sses: <http://sensorlab.ijs.si/2013/v0/SpectrumSensingExperimentSpecification.owl#>
PREFIX om: <http://www.wurvoc.org/vocabularies/om-1.8/>
PREFIX swrlb: <http://www.w3.org/2003/11/swrlb>

SELECT ?Device1 ?Device2 ?Band1 ?Band2
WHERE {
  ?Device1 rdf:type sses:RadioDevice.
  ?Device2 rdf:type sses:RadioDevice.
  ?Device1 sses:supports ?Settings1.
  ?Device2 sses:supports ?Settings2.
  ?Settings1 sses:sweeps ?Band1.
  ?Settings2 sses:sweeps ?Band2.
  ?Band1 sses:startsAt ?f1.
  ?Band2 sses:startsAt ?f2.
  ?Band1 sses:endsAt ?f3.
  ?Band2 sses:endsAt ?f4.
  ?f1 sses:numericalValue ?x.
  ?f2 sses:numericalValue ?y.
  ?f3 sses:numericalValue ?z.
  ?f4 sses:numericalValue ?t.
  FILTER(xsd:integer(?x) = xsd:integer(?y)).
  FILTER(xsd:integer(?z) = xsd:integer(?t)).
}
```

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sses: <http://sensorlab.ijs.si/2013/v0/SpectrumSensingExperimentSpecification.owl#>
PREFIX om: <http://www.wurvoc.org/vocabularies/om-1.8/>
PREFIX swrlb: <http://www.w3.org/2003/11/swrlb>

SELECT ?Device1 ?Device2
WHERE {
  ?Device1 rdf:type sses:RadioDevice.
  ?Device2 rdf:type sses:RadioDevice.
  ?Device1 sses:supports ?Settings1.
  ?Device2 sses:supports ?Settings2.
  ?Settings1 sses:sweeps ?Band1.
  ?Settings2 sses:sweeps ?Band2.
  ?Band1 sses:sameBand ?Band2.
}
```

```
Rules:
Rules +
AntennaFrequencyBand(?B1),
AntennaFrequencyBand(?B2),
endsAt(?B1, ?f2), endsAt(?B2,
?f4), startsAt(?B1, ?f1),
startsAt(?B2, ?f3),
numericalValue(?f1, ?x),
numericalValue(?f2, ?y),
numericalValue(?f3, ?z),
numericalValue(?f4, ?t),
equal(?x, ?z), equal(?y, ?t) ->
sameBand(?B1, ?B2)
```



Complex SPARQL Query: Find all spectrum sensing devices capable of sensing frequency X

Search in Tasor

Execute search

Execution Time: 81596ms.

Execute SPARQL on repository: <http://www.wurvoc.org/vocabularies/om-1.8/>

```
SELECT distinct ?band ?band_prefLabel
WHERE {
  { SELECT ?multIN WHERE {
    BIND(xsd:float("2412") as ?valueINstr)
    BIND("MHz" as ?unitINstr)

    ?unitIN <http://www.wurvoc.org/vocabularies/om-1.8/dimension>
    <http://www.wurvoc.org/vocabularies/om-1.8/frequency_dimension> .
    ?unitIN <http://www.wurvoc.org/vocabularies/om-1.8/symbol> ?s .
    # FILTER (lcase(str(?s)) = lcase(str(?unitINstr)) && substr(str(?s),1,1) =
    substr(str(?unitINstr),1,1))
    FILTER (str(?s) = str(?unitINstr))
    OPTIONAL {
      ?unitIN <http://www.wurvoc.org/vocabularies/om-1.8/prefix> ?prefixIN .
      ?prefixIN <http://www.wurvoc.org/vocabularies/om-1.8/factor> ?scalefactorIN .
    }
    BIND (if( Bound(?scalefactorIN), xsd:float(?scalefactorIN),
    1)*xsd:float(?valueINstr) as ?multIN)
  }}

# <http://sensorlab.ijs.si/2013/v0
/SpectrumSensingExperimentSpecification.owl#startsAt> <http://www.w3.org
/2000/01/rdf-schema#domain> ?bandtypebase .
# ?bandtype rdfs:subClassOf* ?bandtypebase .
```

Search query results here...
RF SBX band
UHF range
2412-2482 MHz
2412-2472 MHz
2400-2495 MHz
2400-2485 MHz
2400-2500 MHz
2Hz-8GHz
2402-2480 MHz
2402-2478 MHz

Frequency Value: 2412

Frequency Unit: MHz

**Results:
List of device capabilities
that cover 2.412 GHz**

Summary

- We have defined and developed a working Spectrum Sensing Ontology that is openly available
- We have modelled several devices and exemplified on use cases how it can be use
- There are several other aspects related to the spectrum sensing device and experiment specification that we have identified but are not yet solved in the current version

- Questions?
- I am happy to provide further explanations and show the running demos locally in Protégé or remotely in TaasOR (today and next days here at FIA)
- Thank you!
- <http://sensorlab.ijs.si/2013/v0/SpectrumSensingExperimentSpecification.owl>