

Tips for Implementing OSLC Adapters

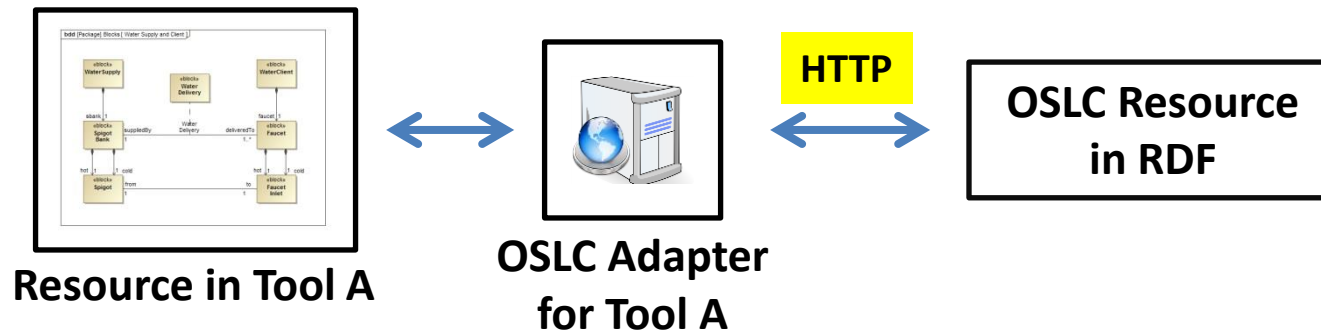
Axel Reichwein

March 24, 2016

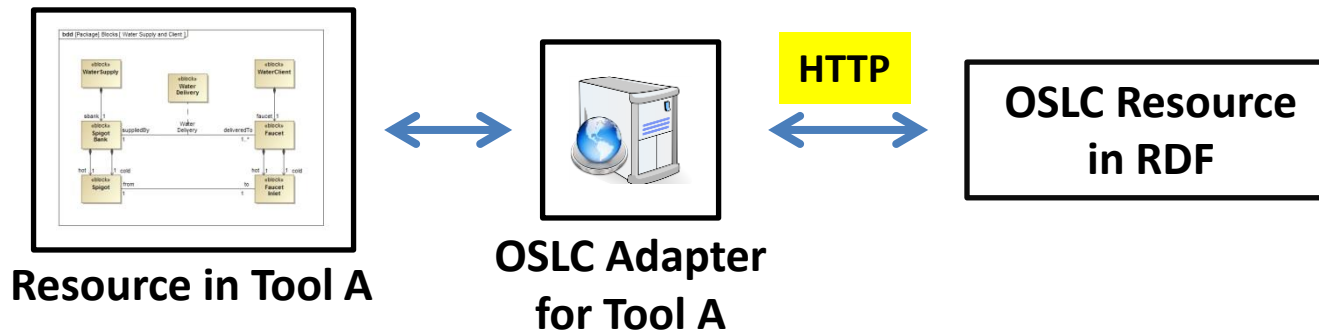
Model-driven Development of OSLC Adapters

- Automatically creating code and RDF resources based on a common domain-specific metamodel
 - Java classes with OSLC4J annotations for serializing POJOs into RDF/XML, JSON
 - Domain-specific API to parse XMI models
 - OSLC resource shapes
 - RDF Vocabulary

General Overview of OSLC Adapter



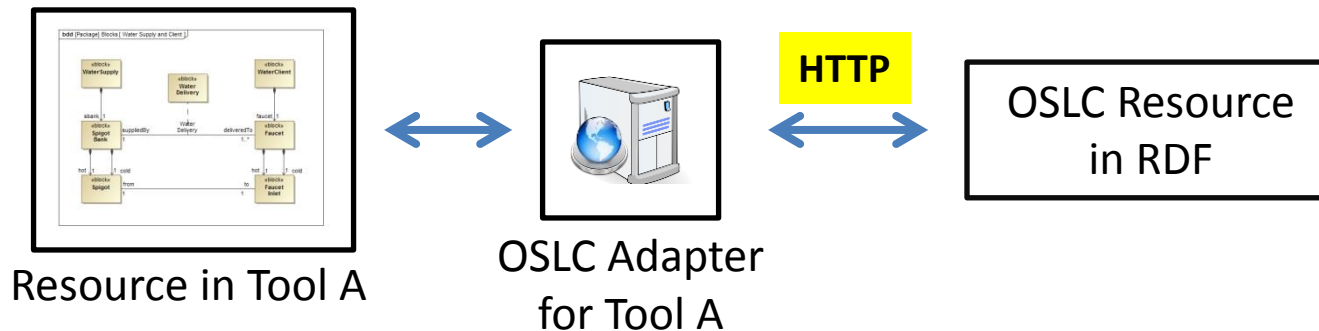
OSLC Adapter Implemented in Java



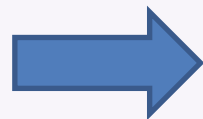
RESTful Web Services of OSLC adapter implemented in Java using **JAX-RS** (Java API for RESTful Web Services) as in the OSLC adapters available from Eclipse Lyo

Converting Java Objects into OSLC Resources in RDF/XML

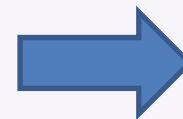
Assumption: Tool has Java API



**Tool-specific
Java objects**



**Java objects
with OSLC
annotations**



**OSLC
Resources in
RDF/XML or
JSON**

Conversion by OSLC adapter

*Automatic conversion from POJO
to RDF/XML or JSON by **OSLC4J**
Annotation Processing Tool*

Example OSLC Resource in RDF/XML

OSLC Resource in RDF/XML With URI:

<http://localhost:8181/oslc4jsimulink/services/model11/blocks/Step>

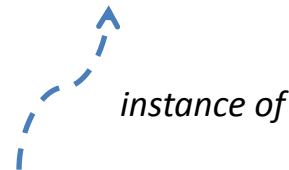
```
<rdf:Description rdf:about="http://localhost:8181/oslc4jsimulink/services/model11/blocks/Step">
  <simulink block:parameter rdf:resource="
http://localhost:8181/oslc4jsimulink/services/model11/parameters/Step::SampleTime"/>
  <simulink block:type rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral
">Step</simulink block:type>
  <simulink block:parameter rdf:resource="
http://localhost:8181/oslc4jsimulink/services/model11/parameters/Step::ZeroCross"/>
  <simulink block:parameter rdf:resource="
http://localhost:8181/oslc4jsimulink/services/model11/parameters/Step::VectorParams1D"/>
  <simulink block:outputPort rdf:resource="
http://localhost:8181/oslc4jsimulink/services/model11/outputports/Step::outport::1"/>
  <rdf:type rdf:resource="http://mathworks.com/simulink/rdf#Block"/>
  <simulink block:parameter rdf:resource="
http://localhost:8181/oslc4jsimulink/services/model11/parameters/Step::After"/>
  <simulink block:parameter rdf:resource="
http://localhost:8181/oslc4jsimulink/services/model11/parameters/Step::Time"/>
  <simulink block:parameter rdf:resource="
http://localhost:8181/oslc4jsimulink/services/model11/parameters/Step::Before"/>
  <simulink block:name rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral
">Step</simulink block:name>
```

Just a snippet

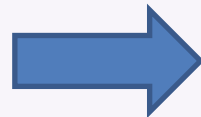
OSLC4J Example

```
@OslcNamespace(Constants.SIMULINK_NAMESPACE)
@OslcName("Block")
@OslcResourceShape(title = "Block Resource Shape", descri
public class SimulinkBlock extends AbstractResource{
```

OSLC Java classes

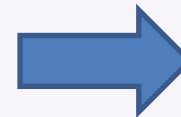


**Tool-specific
Java objects**



Conversion by OSLC adapter

**Java objects
with OSLC
annotations**



**OSLC
Resources in
RDF/XML or
JSON**

*Automatic conversion from POJO
to RDF/XML or JSON by **OSLC4J**
Annotation Processing Tool*

Example Java Class with OSLC4J Annotations

```
@OslcNamespace(Constants.SIMULINK_NAMESPACE)
@OslcName("Block")
@OslcResourceShape(title = "Block Resource Shape", describes = Constants.TYPE_SIMULINK_BLOCK)
public class SimulinkBlock extends AbstractResource{

    public SimulinkBlock() throws URISyntaxException {
        super();
    }
    public SimulinkBlock(URI about) throws URISyntaxException {
        super(about);
    }

    private String name;

    public void setName(String name) {
        this.name = name;
    }

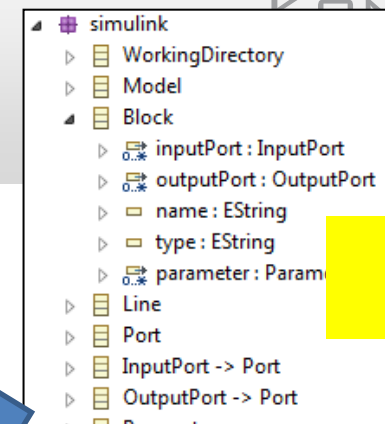
    @OslcDescription("Description of Block::name TBD")
    @OslcName("name")
    @OslcOccurs(Occurs.ZeroOrOne)
    @OslcPropertyDefinition("http://mathworks.com/simulink/rdf#Block/name")
    @OslcTitle("name")
    @OslcValueType(ValueType.XMLLiteral)
    public String getName() {
        return name;
    }
    private String type;

    public void setType(String type) {
        this.type = type;
    }

    @OslcDescription("Description of Block::type TBD")
    @OslcName("type")
    @OslcOccurs(Occurs.ZeroOrOne)
    @OslcPropertyDefinition("http://mathworks.com/simulink/rdf#Block/type")
    @OslcTitle("type")
    @OslcValueType(ValueType.XMLLiteral)
    public String getType() {
        return type;
    }
}
```

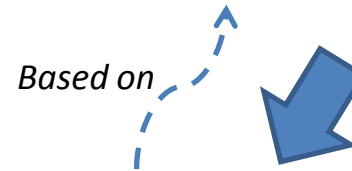
Manually
programming OSLC
Java classes is time-
consuming and
error-prone

Model-driven Generation of OSLC-annotated Java Classes



Frequent changes

Metamodel



Automatic Generation

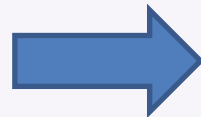
```
@OslcNamespace(Constants.SIMULINK_NAMESPACE)
@OslcName("Block")
@OslcResourceShape(title = "Block Resource Shape", descri
public class SimulinkBlock extends AbstractResource{
```

OSLC-annotated Java classes

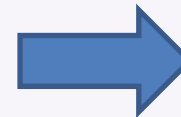
Java classes need to be updated based on metamodel



Tool-specific
Java objects



Java objects
with OSLC
annotations



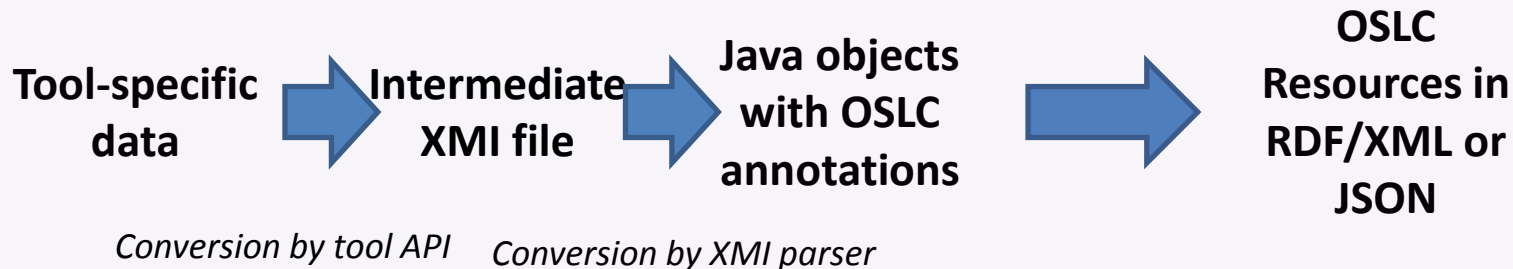
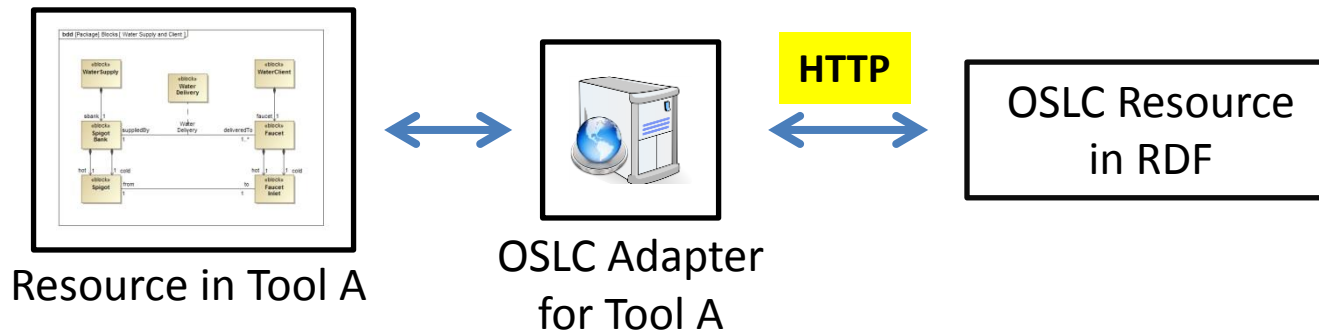
OSLC
Resources in
RDF/XML or
JSON

Conversion by OSLC adapter

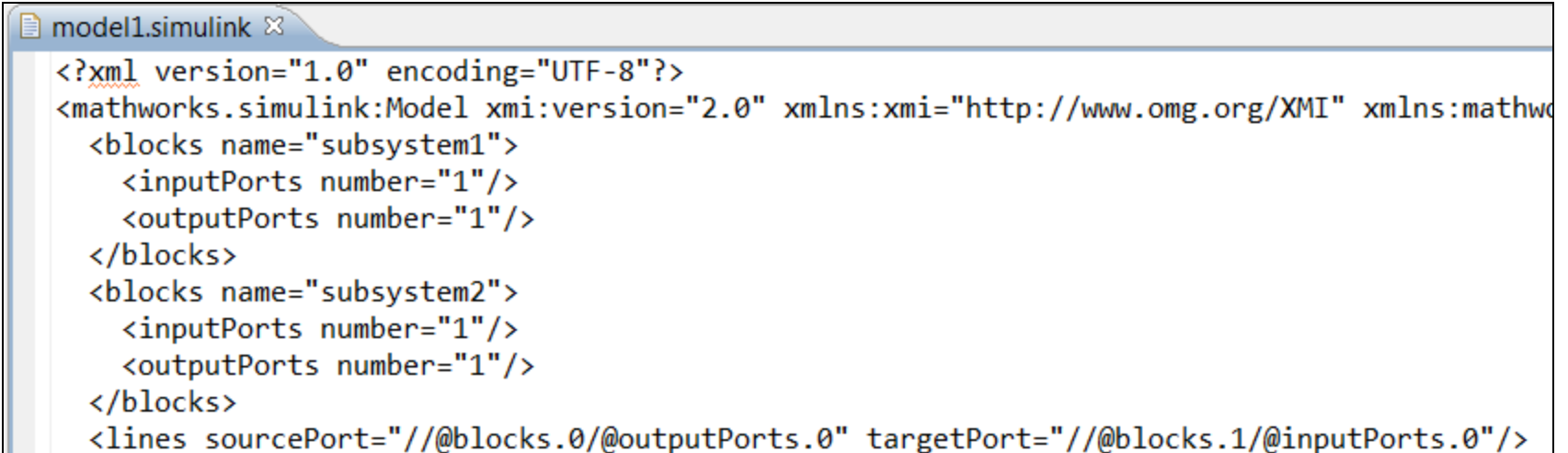
*Automatic conversion from POJO
to RDF/XML or JSON by **OSLC4J**
Annotation Processing Tool*

Converting Java Objects into OSLC Resources in RDF/XML

Assumption: Tool has NO Java API



Example Intermediate XMI File

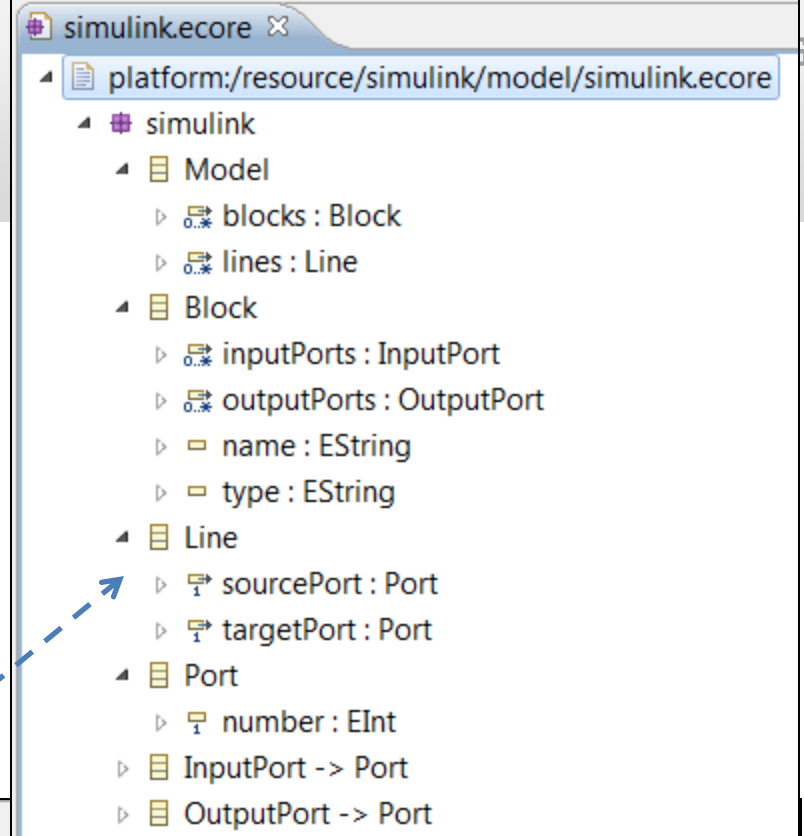


```
<?xml version="1.0" encoding="UTF-8"?>
<mathworks:simulink:Model xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:mathwo
  <blocks name="subsystem1">
    <inputPorts number="1"/>
    <outputPorts number="1"/>
  </blocks>
  <blocks name="subsystem2">
    <inputPorts number="1"/>
    <outputPorts number="1"/>
  </blocks>
  <lines sourcePort="//@blocks.0/@outputPorts.0" targetPort="//@blocks.1/@inputPorts.0"/>
```

XML Schema of XMI File

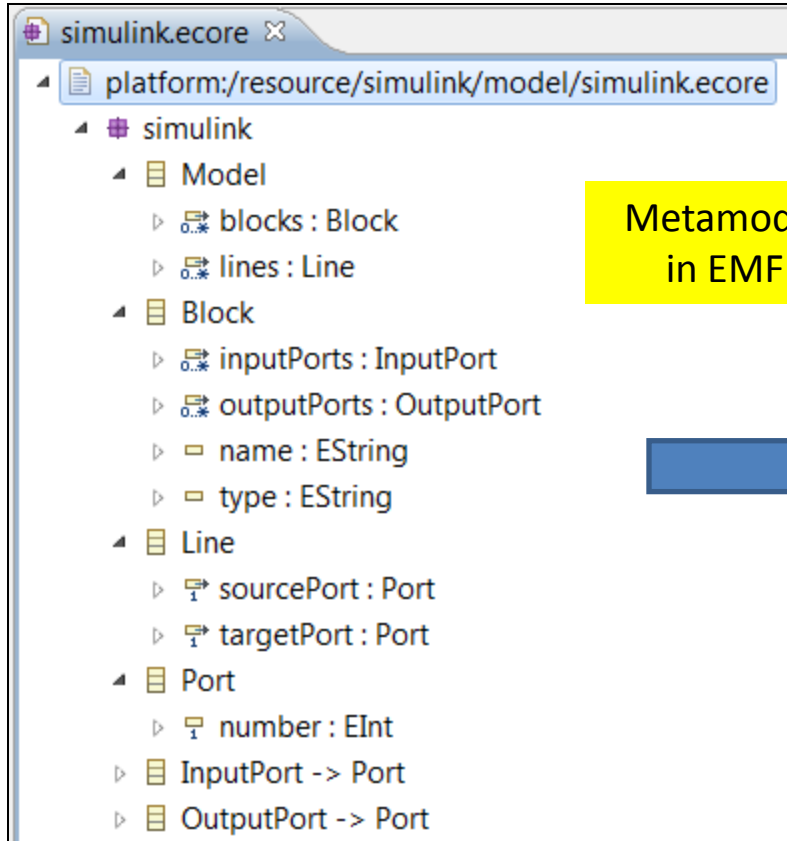
- XMI file needs to be parsed easily!
- XMI file parser can be generated with EMF based on Simulink Ecore metamodel
- XML Schema of XMI File defined as Ecore metamodel

conformsTo



```
model1.simulink x
<?xml version="1.0" encoding="UTF-8"?>
<mathworks:simulink:Model xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:mathw
  <blocks name="subsystem1">
    <inputPorts number="1"/>
    <outputPorts number="1"/>
  </blocks>
  <blocks name="subsystem2">
    <inputPorts number="1"/>
    <outputPorts number="1"/>
  </blocks>
  <lines sourcePort="//@blocks.0/@outputPorts.0" targetPort="//@blocks.1/@inputPorts.0"/>
```

Generation of Domain-specific API



Metamodel
in EMF

```
<?xml version="1.0" encoding="UTF-8"?>
<mathworks:simulink:Model xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:mathw...
  <blocks name="subsystem1">
    <inputPorts number="1"/>
    <outputPorts number="1"/>
  </blocks>
  <blocks name="subsystem2">
    <inputPorts number="1"/>
    <outputPorts number="1"/>
  </blocks>
  <lines sourcePort="//@blocks.0/@outputPorts.0" targetPort="//@blocks.1/@inputPorts.0"/>
</mathworks:simulink:Model>
```

Generation of Java code to
parse corresponding XMI file

```
public class SimulinkModelParser {

    public static void main(String[] args) {

        // load Simulink file
        Resource ecoreResource = LoadEcoreModel(URI.createFileURI(new File(
            "model1.simulink").getAbsolutePath()));

        // load Simulink model
        Model simulinkModel = (Model) EcoreUtil.getObjectByType(
            ecoreResource.getContents(),
            SimulinkPackage.eINSTANCE.getModel());

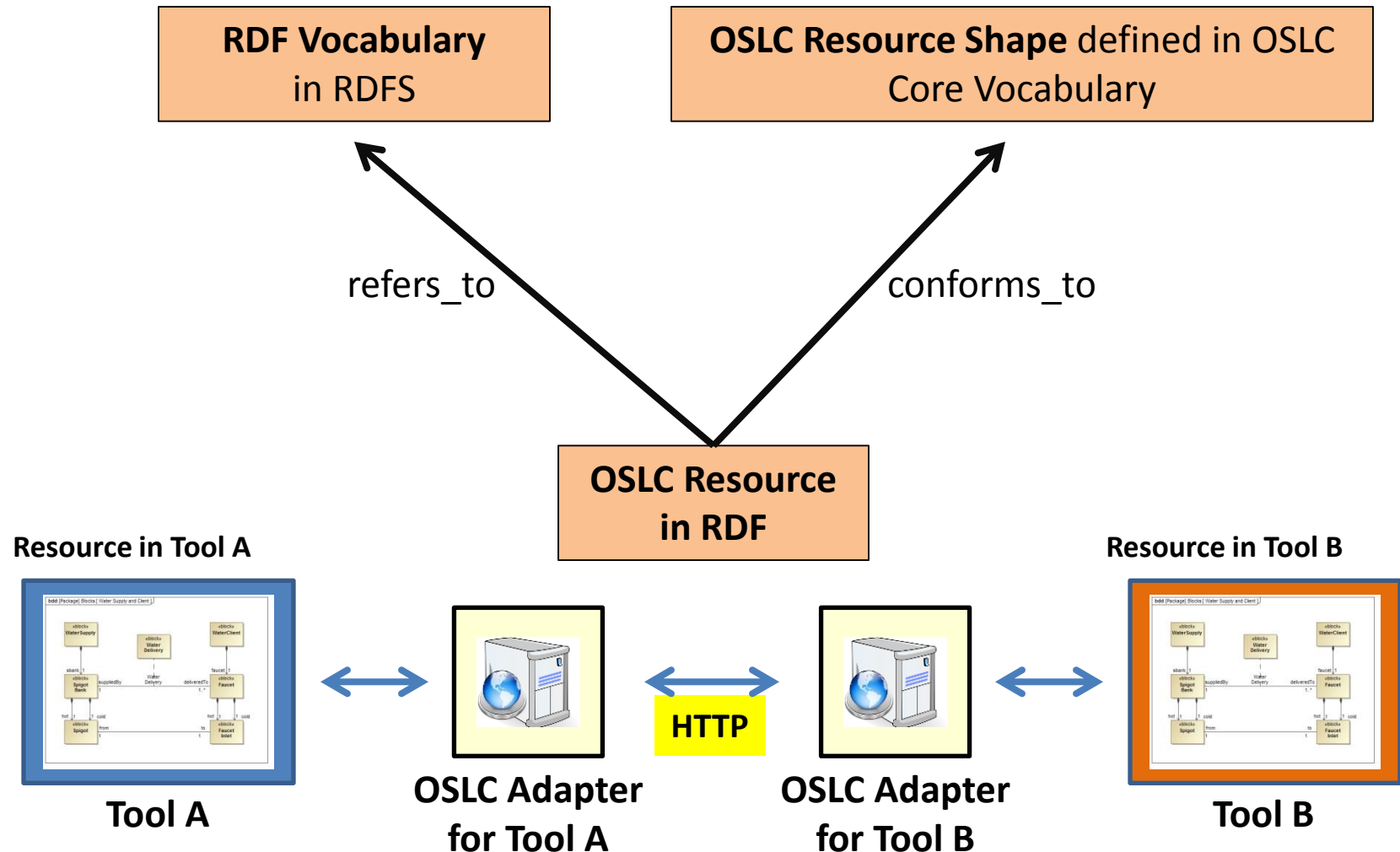
        // print block names
        for (Block block : simulinkModel.getBlocks()) {
            System.out.println(block.getName());
        }
    }
}
```

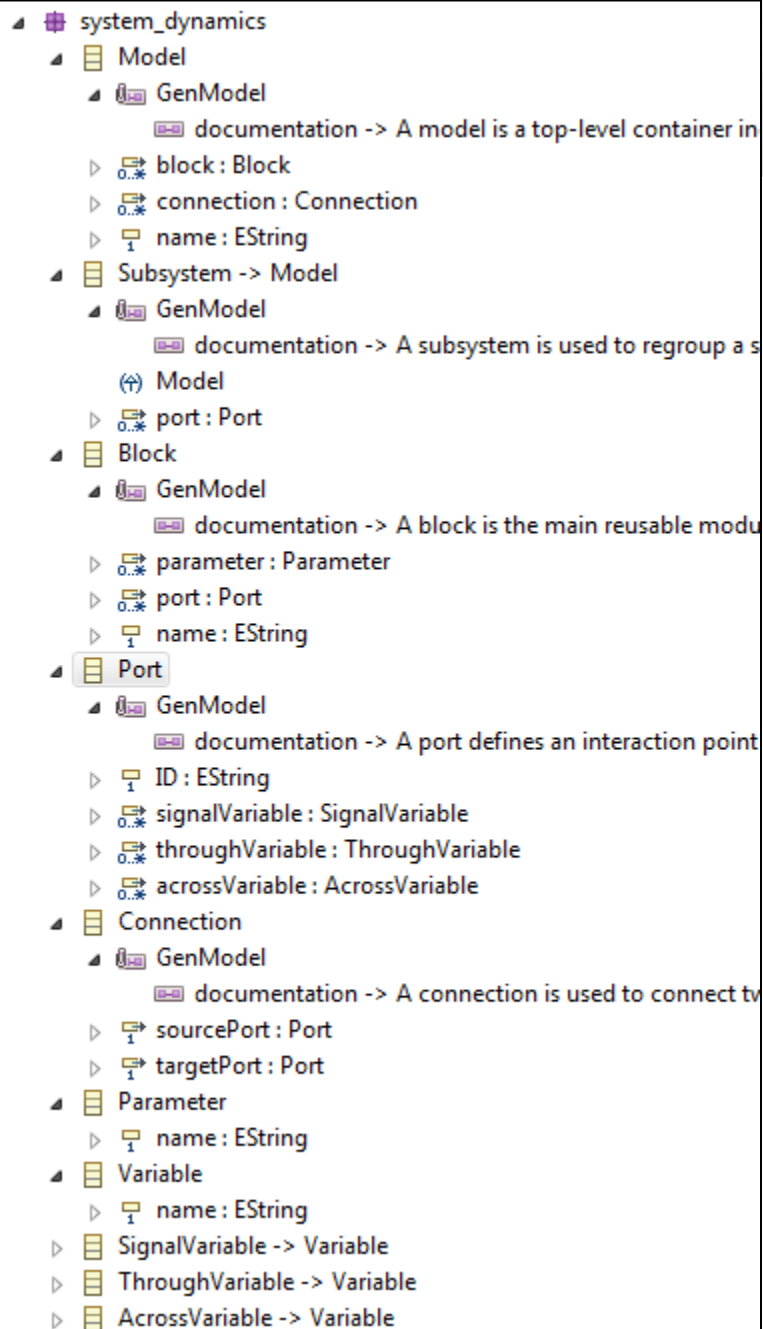
Results

<terminated> SimulinkModelParser [Main] 7.0.13\bin\javaw.exe (19...
subsystem1
subsystem2

General Overview of RDF

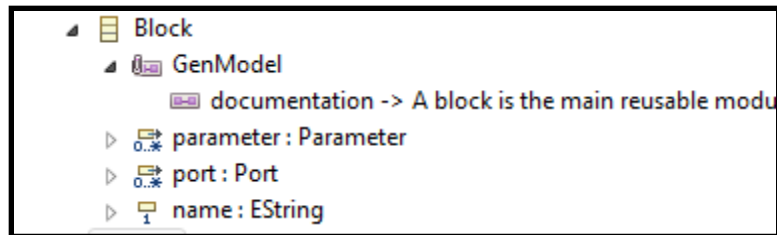
Resources for OSLC Data Interchange





Conversion of Ecore Metamodel to RDF Resources of corresponding OSLC Specification

Conversion of Ecore Metaclass and Features into RDFS Class and RDF properties



Reuse of
Dublin Core
Vocabulary

Namespace URI
of RDF Vocabulary

```
<rdf:Class rdf:about="dynsim:Block">
  <rdf:label xml:lang="en-GB">Block</rdf:label>
  <dcterms:description xml:lang="en-GB">A block is the main reusable modular unit
  for describing a physical system or an operation within a dynamic system model.
  </dcterms:description>
  <rdf:isDefinedBy rdf:resource="http://incose.org/dynsim#" />
  <dcterms:issued>2014-01-05</dcterms:issued>
</rdf:Class>
<rdf:Property rdf:about="dynsim:Block_name">
  <rdf:label xml:lang="en-GB">name</rdf:label>
  <rdf:isDefinedBy rdf:resource="http://incose.org/dynsim#" />
  <dcterms:issued>2014-01-05</dcterms:issued>
</rdf:Property>
<rdf:Property rdf:about="dynsim:Block_parameter">
  <rdf:label xml:lang="en-GB">parameter</rdf:label>
  <rdf:isDefinedBy rdf:resource="http://incose.org/dynsim#" />
  <dcterms:issued>2014-01-05</dcterms:issued>
</rdf:Property>
<rdf:Property rdf:about="dynsim:Block_port">
  <rdf:label xml:lang="en-GB">port</rdf:label>
  <rdf:isDefinedBy rdf:resource="http://incose.org/dynsim#" />
  <dcterms:issued>2014-01-05</dcterms:issued>
</rdf:Property>
```


RDF Vocabulary

for Dynamic Simulation Domain

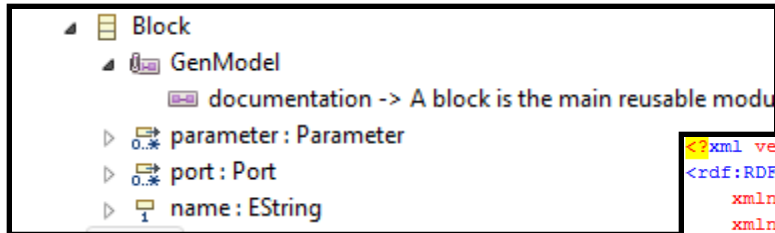
RDF Vocabulary
Namespace
prefix and URI

RDFS Classes
and RDF
Properties are
defined within
an RDF
Vocabulary

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dynsim="http://incose.org/dynsim#"
  <rdfs:Class rdf:about="dynsim:Model">
    <rdfs:label xml:lang="en-GB">Model</rdfs:label>
    <dcterms:description xml:lang="en-GB">A model is a top-level contain
    <rdfs:isDefinedBy rdf:resource="http://incose.org/dynsim#">
    <dcterms:issued>2014-01-05</dcterms:issued>
  </rdfs:Class>
  <rdf:Property rdf:about="dynsim:Model_name">
    <rdfs:label xml:lang="en-GB">name</rdfs:label>
    <rdfs:isDefinedBy rdf:resource="http://incose.org/dynsim#">
    <dcterms:issued>2014-01-05</dcterms:issued>
  </rdf:Property>
  <rdf:Property rdf:about="dynsim:Model_block">
    <rdfs:label xml:lang="en-GB">block</rdfs:label>
    <rdfs:isDefinedBy rdf:resource="http://incose.org/dynsim#">
    <dcterms:issued>2014-01-05</dcterms:issued>
  </rdf:Property>
  <rdf:Property rdf:about="dynsim:Model_connection">
    <rdfs:label xml:lang="en-GB">connection</rdfs:label>
    <rdfs:isDefinedBy rdf:resource="http://incose.org/dynsim#">
    <dcterms:issued>2014-01-05</dcterms:issued>
  </rdf:Property>
  <rdfs:Class rdf:about="dynsim:Subsystem">
    <rdfs:label xml:lang="en-GB">Subsystem</rdfs:label>
    <dcterms:description xml:lang="en-GB">A subsystem is used to regroup
    <rdfs:isDefinedBy rdf:resource="http://incose.org/dynsim#">
    <dcterms:issued>2014-01-05</dcterms:issued>
    <rdfs:subClassOf rdf:resource="dynsim:Model"/>
  </rdfs:Class>
```

Just a snippet...

Conversion of Ecore Metaclass and Features into OSLC Resource Shape



Property names and multiplicities are converted into OSLC resource shape constraints



```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:oslc="http://open-services.net/ns/core#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dynsim="http://incose.org/dynsim#"
  <oslc:ResourceShape rdf:about="http://myOSLCServiceProvider.com/dynsim/BlockResourceShape">
    <oslc:describes rdf:resource="dynsim:Block"/>
    <dcterms:title rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
      Block Resource Shape</dcterms:title>
    <oslc:property>
      <oslc:Property>
        <oslc:name>name</oslc:name>
        <oslc:propertyDefinition rdf:resource="dynsim:Block_name"/>
        <oslc:occurs rdf:resource="http://open-service.net/ns/core#Exactly-one"/>
      </oslc:Property>
    </oslc:property>
    <oslc:property>
      <oslc:Property>
        <oslc:name>parameter</oslc:name>
        <oslc:propertyDefinition rdf:resource="dynsim:Block_parameter"/>
        <oslc:range rdf:resource="dynsim:Parameter"/>
        <oslc:valueType rdf:resource="http://open-services.net/ns/core#Resource"/>
        <oslc:occurs rdf:resource="http://open-service.net/ns/core#Zero-or-many"/>
      </oslc:Property>
    </oslc:property>
    <oslc:property>
      <oslc:Property>
        <oslc:name>port</oslc:name>
        <oslc:propertyDefinition rdf:resource="dynsim:Block_port"/>
        <oslc:range rdf:resource="dynsim:Port"/>
        <oslc:valueType rdf:resource="http://open-services.net/ns/core#Resource"/>
        <oslc:occurs rdf:resource="http://open-service.net/ns/core#Zero-or-many"/>
      </oslc:Property>
    </oslc:property>
  </oslc:ResourceShape>
</rdf:RDF>
```