

Resource Profiles Markup Language



**Stephen Downes
and Luc Belliveau**
Atlantic Workshop
on Semantics and
Services
**Fredericton, NB,
June 14, 2010**

Overview

- Resource Profiles
 - Distributed Metadata
 - Three Types of Metadata
 - Educational Metadata
 - Using Resource Profiles
- Resource Profiles Markup Language
 - RPML Tool Demo
 - RPML Syntax Overview
 - RPML Application Detail

Distributed Metadata

- One Document, One Record?
- Distributed Metadata:
 - Multiple Metadata Formats
 - Multiple Authorship
 - Multiple Locations

Three Types of Metadata

- First Party Metadata
 - Bibliographic, technical, rights
- Second Party Metadata
 - Page rank, usage metadata, server logs, context
- Third Party Metadata
 - Classification, ratings, educational

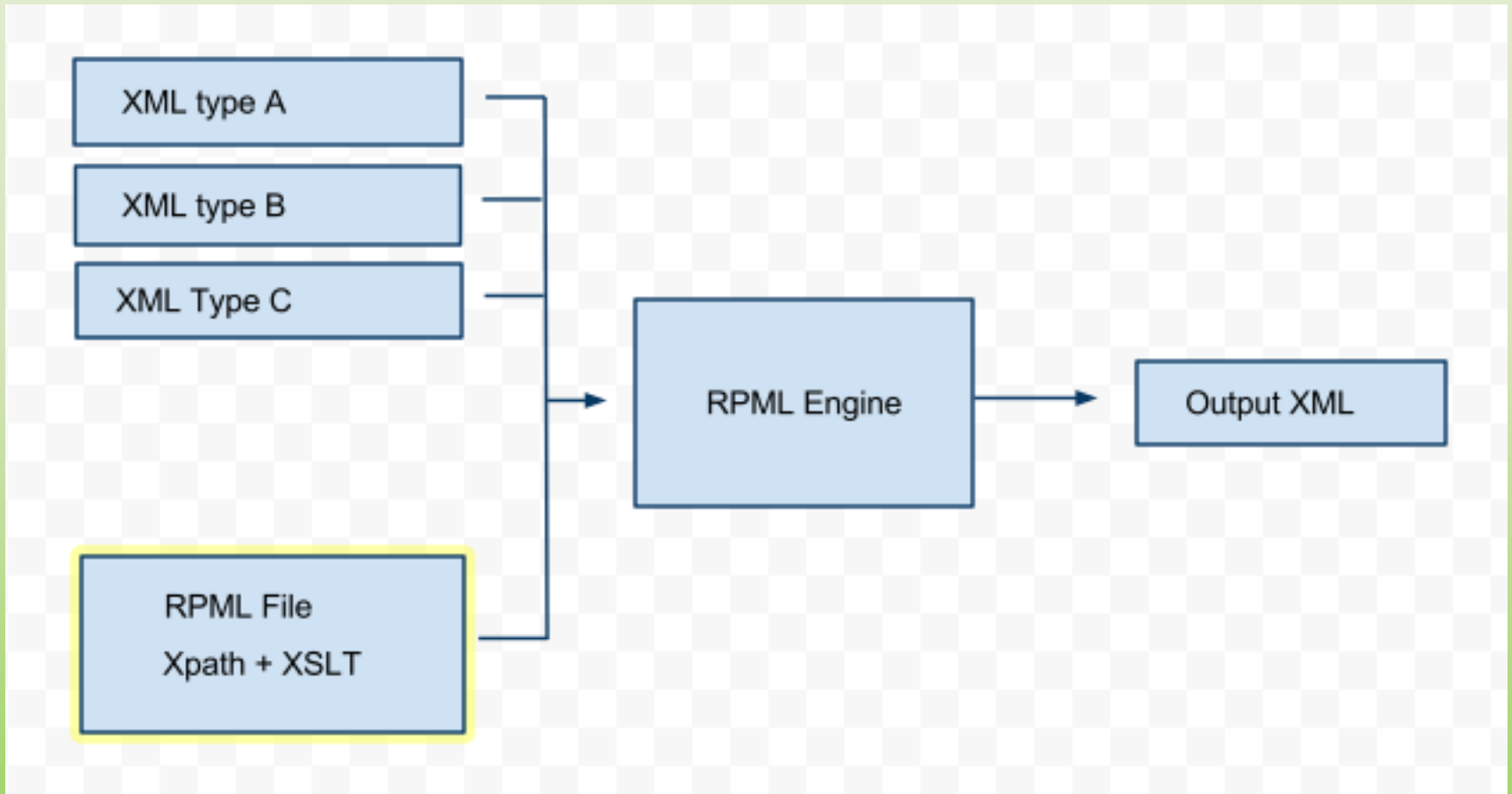
Educational Metadata

- Educational Standards Metadata
 - Matches a resource to an educational standard
 - Curriculum, course description, competencies
- Educational Properties Metadata
 - interactivity type, learning resource type, interactivity level, semantic density
- Educational Use Metadata
 - Relation, sequencing, learning design

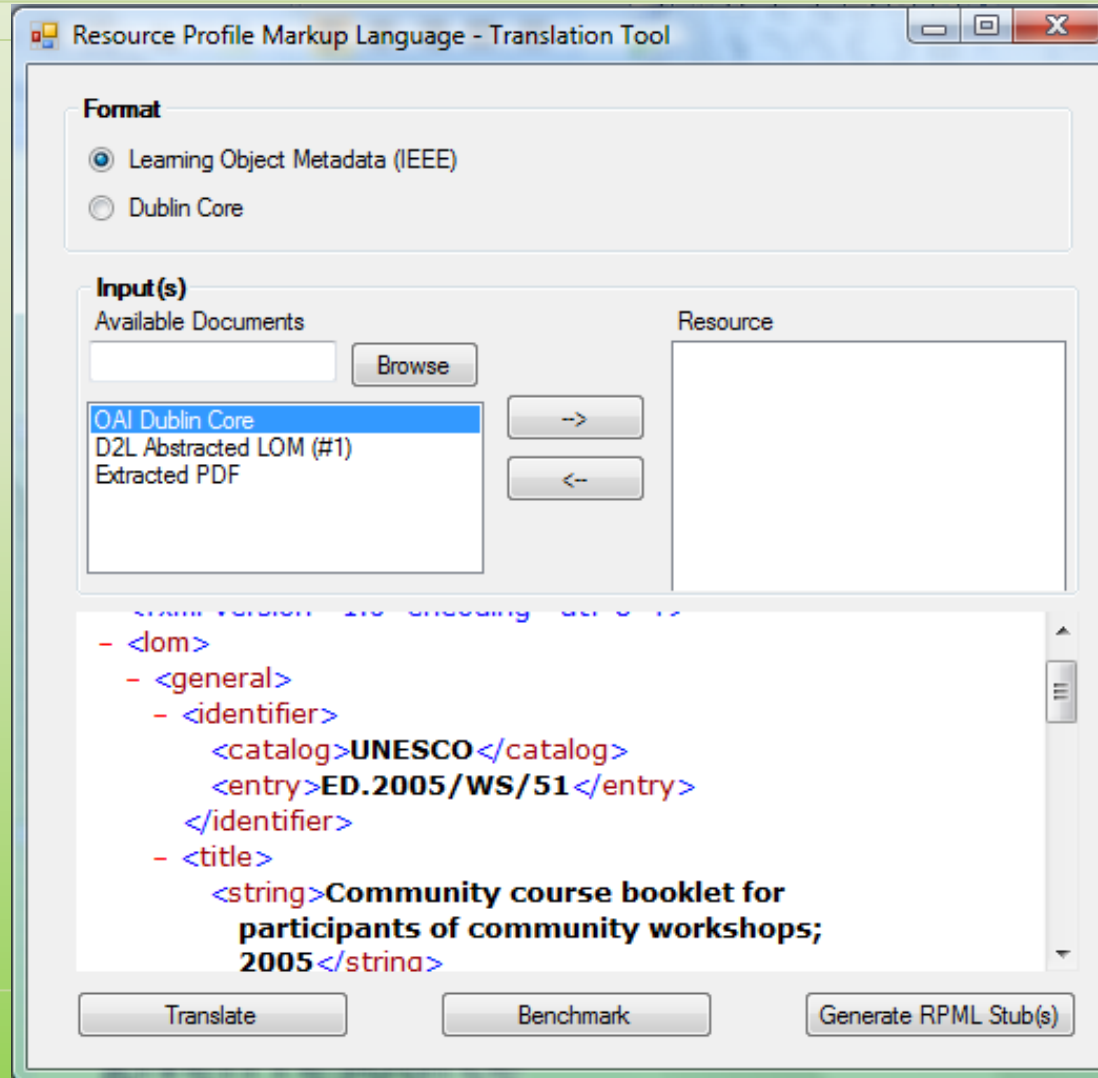
Using Resource Profiles

- Static vs Evolving Resource Description
- Tailoring of Profile to Specific Needs
- Resource Metadata Lifecycle
- The RPML engine selects elements from different files and combines them to form new, and possibly unique, descriptions of a given resource

RPML Engine Demo



RPML Engine Demo



RPML Syntax Overview

Getting Results

```
<directive>
```

```
  <ruleset>
```

```
    <rule test="/dc/title">
```

```
      <add-result name="title" select="/dc/title" />
```

```
    </rule>
```

```
    <rule test="/lom/general/title">
```

```
      <add-result name="title" select="/lom/general/title" />
```

```
    </rule>
```

```
  </ruleset>
```

RPML Syntax Overview 2

Using Results

```
<evaluate>
```

```
  <true>
```

```
    <for-each result="title" select="node()">
```

```
      <element name="title">
```

```
        <value-of select="text()" />
```

```
      </element>
```

```
    </for-each>
```

```
  </true>
```

```
  <>false> <!-- none of the rules matched --!></false>
```

```
</evaluate>
```

```
</directive>
```

RPML App Flow XML

Inputs Xpath

- <inputs>

- <document>

<text>**Extracted PDF**</text>

<uri>**file:///{\$appPath}/examples/inputPdfExtracted.xml**</uri>

</document>

- <document>

<text>**OAI Dublin Core**</text>

<uri>**file:///{\$appPath}/examples/inputOaiDc.xml**</uri>

</document>

</inputs>

</rpmTool>

RPML App Flow XML 2

Outputs XSLT

- <outputs>

- <document>

<text>**Learning Object Metadata (IEEE)**</text>

<uri>**file:///{\$appPath}/examples/outputLom.xml**</uri>
>

</document>

- <document>

<text>**Dublin Core**</text>

<uri>**file:///{\$appPath}/examples/outputDc.xml**</uri>

</document>

</outputs>

RPML XSLT

Rule – Relation Construction of Output File

```
<rpml:first-resource match="rule">  
<rpml:rule test="/oai_dc:dc/dc:relation">  
  <rpml:set-value mode="relation"  
    select="/oai_dc:dc/dc:relation" />  
</rpml:rule>  
<rpml:element name="relation">  
  <rpml:get-value mode="relation" select="text()"  
    />  
</rpml:element>  
</rpml:first-resource>
```

RPML PHP1

```
// Include OAI DC Metadata, a yes/no value and PdfExtract metadata.
```

```
$resources = array(  
    pathinfo('file:///'. $_SERVER['SCRIPT_FILENAME'],  
        PATHINFO_DIRNAME) . '/data/lvS7/md_1',  
    pathinfo('file:///'. $_SERVER['SCRIPT_FILENAME'],  
        PATHINFO_DIRNAME) . '/data/lvS7/md_2',  
    pathinfo('file:///'. $_SERVER['SCRIPT_FILENAME'],  
        PATHINFO_DIRNAME) . '/data/lvS7/md_3'  
);
```

```
// Instantiate the RPML processor
```

```
$rpml = new rpml($resources);
```

```
// Output resulting XML file.
```

```
echo $rpml->getOutput();
```

RPML PHP2

```
$selectedDoc = new DOMDocument();  
$xslDoc = new DOMDocument();  
$xsl = new XSLTProcessor();  
$selectedDoc->load(pathinfo.../profile/example.xml');  
$xslDoc->load(pathinfo... ./inc/rpml.xsl');
```

```
    $xsl->importStylesheet($xslDoc);  
    $resources = implode(',', $this->resources);  
    $xsl->setParameter('', 'resources', $resources);  
    $data = '';  
    $data = $xsl->transformToXml($selectedDoc);
```

Thanks!

Stephen Downes

stephen@downes.ca

Luc Belliveau

Luc.Belliveau@nrc-cnrc.gc.ca