Dec 14, 2004

Agenda

Igor's comments

Attendees

Darren
Fred
Andreas
Ali
Steve
Igor
Michel
Akioka

[Text Slide A]

Igor's Comments

##
The is old and this is already been replaced.
The [<jsdl:Extend name="PBSQueueName">myQueue</jsdl:Extend>] should actually be
[<pbs:QueueName>myQueue</pbs:QueueName>].

If it stays this way, JSDL will be beaten for improper XML extensibility approach. Could you please forward my comment to JSDL list?
##

Use of QNames and enumerations

Do we need closed and opened sets for enumerations?

One can't do this:

[

<xsd:simpleType name="frequencyUnits">
<xsd:restriction base="xsd:QName">
<xsd:enumeration value="Hz"/>
<xsd:enumeration value="kHz"/>
<xsd:enumeration value="MHz"/>
<xsd:enumeration value="GHz"/>
<xsd:enumeration value="THz"/>

<xsd:enumeration value="PHz"/>
<xsd:enumeration value="EHz"/>
</xsd:restriction>
</xsd:simpleType>

]

This is not a QName enumeration this is a string enumeration!! QName MUST be qualified and it MUST be decalred somewhere.

And another: [<xsd:union memberTypes="xsd:QName">] Most tools will choke on this.

[Text Slide B]

Enumeration
First Option:

The general extension mechanism is used.
<CadenceProcessor>foo</CadenceProcessor>

Open Enumerations
Second Option:

```
<ProcessorArchitecture>
  <Pentium/>
</ProcessorArchitecture>

<ProcessorArchitecture>
  <Cadance:SpecialProcessor/>
</ProcessorArchitecture>

<ProcessorArchitecture>
  <CIM:ProcessorType value="4543"/>
</ProcessorArchitecture>

##################################
# This is the best of the choices
<ProcessorArchitecture>
  <ProcessorArchitectureName>Pentium</ProcessorArchitectureName>
</ProcessorArchitecture>
```

```
<ProcessorArchitecture>
  <ProcessorArchitectureName>Pentium</ProcessorArchitectureName>
  <CIM:ProcessorType>11</CIM:ProcessorArch>
</ProcessorArchitecture>
```

In last example the JSDL consumer MAY choose to ignore the extension (<CIM:ProcessorType>) element in every case except where the ProcessorArchitectureName is "other".

```
<ProcessorArchitecture>
  <ProcessorArchitectureName>Spark</ProcessorArchitectureName>
  <CIM:ProcessorType>11</CIM:ProcessorArch>
</ProcessorArchitecture>
```

This last example is invalid - as JSDL says Spark and CIM says Pentium
However only if the consumer chooses not to ignore the extension should this be considered as an error.

```
<ProcessorArchitecture>
  <ProcessorArchitectureName>other</ProcessorArchitectureName>
  <CIM:ProcessorType>23</CIM:ProcessorArch>
</ProcessorArchitecture>
```

```
<xsd:element name="CPUArchitecture">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:element name="CPUArchitectureName" type="jsdl:ProcessorArchitectureEnumeration" />
   <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
```

```
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>
</xsd:element>

##################################

<jsdl:ProcArch>...</..>
<cim:ProcArch>...</...>
```

[Text Slide C]

<xs:schema elementFormDefault="qualified" xmlns:tns="...">

<xs:element name="abc" type="tns:def"/>

This should be done to the schema.

<xs:complexType name="xyz">
<xs:complexContent>
<xs:sequence>
....
<xs:any .../>
</xs:sequence>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexContent>
</xs:complexType>

This should be added to ALL complex type.

[Text Slide D]

[Text Slide E]