PGI Requirement 110 Clarification
Mark Morgan

I was asked to clarify range types as they appear in the JSDL specification. After looking through the section on ranges in the JSDL document, I couldn't see anything that I felt was unclear and so I decided to look at this assignment as an opportunity to explain what was already there rather than attempt a clarification. To that end, I have cut-and-paste the original text from the JSDL document and will comment "inline" on the text in an attempt to explain the types.

For the purposes of both the JSDL and of my explanations, the standard mathematical notation of square brackets ("[", "]") are used to indicate respectively an inclusive lower, and inclusive upper boundary on a range, and the parenthesis "(", and ")" respectively indicate exclusive range boundaries on the lower and upper end. Therefore, the expression "[10, 20)" would indicate a range starting at 10 and ending with 19 (assuming that the set of possible numbers was the integers).

That text follows:

5.2.5 RangeValue_Type Type

A range value is a complex type that allows the definition of exact values (with an optional "epsilon" argument), left-open or right-open intervals and ranges. All numbers given are of type xsd:double. UpperBoundedRanges and LowerBoundedRanges are limited to the upper or lower bound, respectively. Ranges may be "unlimited" to either negative or positive infinity, subject to the consuming system's capabilities. For example, expressed in Java, an implementation of "infinity" could be java.lang.Double.NEGATIVE_INFINITY and java.lang.Double.POSITIVE_INFINITY, respectively.

Mark Morgan 6/29/10 9:15 AM
Comment: All values in JSDL that need to be expressed as a "possible" range have this schema type.

Mark Morgan 6/29/10 9:16 AM
Comment: Range types can be exact matches (with an epsilon), or inclusive or exclusive ranges possible unbounded on one end or the other of the real number line.

The optional attribute "exclusiveBound" has the default value of "false" if not specified.

If the optional attribute "epsilon" is not specified it defaults to the value of 0 and an exact match up to the precision available on the consuming system MUST be carried out by the consuming system. If the "epsilon" attribute is specified, then its value determines the allowed imprecision when testing a given value against the value of the "Exact" element. Hence an "Exact" element with a present "epsilon" attribute value implicitly defines a Range "[value-epsilon, value+epsilon]". The value of the "epsilon" attribute MUST NOT be negative.

Mark Morgan 6/29/10 9:16 AM
Comment: Unless otherwise specified, ranges use inclusive bounds.

RangeValues that specify intersecting ranges MAY be collapsed by the consuming system in order to match the given job description, but the JSDL document MUST NOT be changed.

Mark Morgan 6/29/10 9:24 AM
Comment: A system consuming JSDL may choose to collapse overlapping ranges for processing, but should not modify the document permanently.

This type MUST be supported by the consuming system. The formal matching semantics are de- fined in the next section (§5.2.5.1).

5.2.5.1 Matching Semantics

The following matching semantics MUST be used.

- A numeric value, *N*, matches a RangeValue, *R*, if and only if at least one of the following conditions is true:
  - *R* contains an UpperBoundedRange, *U*, with a false exclusiveBound attribute and where $N \leq U$.
  - *R* contains an UpperBoundedRange, *U*, with a true exclusiveBound attribute and where $N < U$.
  - *R* contains a LowerBoundedRange, *L*, with a false exclusiveBound attribute and where $N \geq L$.
  - *R* contains a LowerBoundedRange, *L*, with a true exclusiveBound attribute and

Mark Morgan 6/29/10 9:26 AM
Comment: N is in R if R = (-∞, U] and N ≤ U

Mark Morgan 6/29/10 9:27 AM
Comment: N is in R if R = (-∞, U) and N < U

Mark Morgan 6/29/10 9:28 AM
Comment: N is in R if R = [L, ∞) and N ≥ L

where $N > L$.

- R contains an Exact, E, with an epsilon attribute e, where $E-e \leq N \leq E+e$.
- R contains a Range with LowerBound, L, and UpperBound, U, such that both of the following are true:
  - L has a false exclusiveBound attribute and $N \geq L$, or L has a true exclusiveBound attribute and $N > L$.
  - U has a false exclusiveBound attribute and $N \leq U$, or U has a true exclusiveBound attribute and $N < U$.

Mark Morgan 6/29/10 9:29 AM
Comment: N is in R if R = (L, ∞) and N > L

Mark Morgan 6/29/10 9:32 AM
Comment: N matches E if E = [E-e, E+e] and E-e <= N <= E+e and assuming that e is the epsilon value (possibly e = 0).

Mark Morgan 6/29/10 9:34 AM
Comment: N matches R as per the rules for lower and upper bounds describe above for open ended ranges but in this case only if both the lower bound and upper bound constraints are satisfied.

5.2.5.2 Pseudo Schema

```
<UpperBoundedRange exclusiveBound="xsd:boolean"?>
     xsd:double
</UpperBoundedRange>?
<LowerBoundedRange exclusiveBound="xsd:boolean"?>
     xsd:double
</LowerBoundedRange>?
<Exact epsilon="xsd:double"?>
     xsd:double
</Exact>*
<Range>
     <LowerBound exclusiveBound="xsd:boolean"?>
          xsd:double
     </LowerBound>
     <UpperBound exclusiveBound="xsd:boolean"?>
          xsd:double
     </UpperBound>
</Range>*
```

5.2.5.3 Examples

The pseudo-expression “5, 6.7777, 7.0, [50.3,99.5), [100-” (which indicates a disjoint range consisting of the values 5, 6.7777, 7.0, all values inclusive of from 50.3 up to, but not including, 99.5 and all values from 100 on up) may be encoded in a RangeValue as follows:

```
<jsdl:LowerBoundedRange> 100.0 </jsdl:LowerBoundedRange>
<jsdl:Exact> 5.0 </jsdl:Exact>
<jsdl:Exact epsilon="0.00001"> 6.7777 </jsdl:Exact>
<jsdl:Exact> 7.0 </jsdl:Exact>
<jsdl:Range>
     <jsdl:LowerBound> 50.3 </jsdl:LowerBound>
<jsdl:UpperBound exclusiveBound="true"> 99.5 </jsdl:UpperBound>
</jsdl:Range>
```