DFDL WG Session 3

Mike Beckerle

Ascential Software

- Two note-takers please?

DFDL-WG Session 3
Data Model and Binary Primitives
(The old agenda - from the program)

(If the previous sessions issues still need time they will be worked on in this session.)

- Discussion of issues with primitives set of W3C Schema
  - identify important gaps (multidimensional arrays, for example)
  - identify anomalous semantics
- Discussion of binary mappings
  - list mappings and arguments
  - expose issues
  - propose naming policy/names for mappings
- Discussion of binary mapped types
  - list types
  - naming policy/name
  - Discussion of text mappings
  - list mappings and arguments
  - expose issues
  - propose naming policy/names for mappings
- Discussion of text mapped types
  - list types
  - naming policy/names

DFDL-WG Session 3 Agenda

- Current Working Issues
  - (Continued from Session 2)
- Data Model and Mapping Primitives

Issues (From Session 2)

- Stored length, references in general
- Choice/unions
  - Expression language for discrimination
- Layered translations
  - compression, encryption
  - IBM data streams (F, FB, VB, VBS)
- Modularity
  - How to plug in new transformations?
  - Composition properties

Layered Translations

- Use case: Matrix with dynamic size in text file:
  - blank lines are ignored
  - C-style comments are ignored (equiv. to whitespace)
  - First line contains xdim ydim (whitespace separated)
  - Subsequent lines are rows of the 2-d matrix.
    - There must be exactly ydim rows
      - each containing xdim numbers
    - Within each row the values are whitespace separated.
  - The charset is UTF-8
- Requires that we express preprocessing of the input data to handle the C-style comments and blank lines
- The preprocessing is not part of the structure of the data

Layered Translations
Matrix w/Dynamic Size Example

```
/* obsv3 •€• 2003•08•27•   •€  */

/* gbxx2. 14:02:21 •€•• 8 •
*/
3 2

/*  •••• €• •••••• €•*/
1 2 3 /**/
4 5 /*•••••*/ 6

/*  •€ •••••• */
```

Layered Translations Matrix w/Dynamic Size Example

```
<dims>
  <xdim>3</xdim>
  <ydim>2</ydim>
</dims>
<ydata>
<xdata>1</xdata>
<xdata>2</xdata>
<xdata>3</xdata>
</ydata>
<ydata>
<xdata>4</xdata>
<xdata>5</xdata>
<xdata>6</xdata>
</ydata>
```

Layered Translations Matrix w/Dynamic Size Example

```
<element name="example2">
<sequence>
  <element name="dims">
    <sequence>
      <element name="xdim" type="int"/>
      <element name="ydim" type="int"/>
    </sequence>
  </element>
<!– XSD/XML Issues: XSD has no 2-d array. Also there is no way to
   constrain minOccurs or maxOccurs based on the value of other
   elements of the XML -->
  <element name=“ydata“ minOccurs=0 maxOccurs=“unbounded”>
    <sequence>
      <element name=“xdata” type=“double”
               minOccurs=“0” maxOccurs=“unbounded”/>
    </sequence>
  </element>
</sequence>
</element>
```

Layered Translations Matrix w/Dynamic Size Example

- Underlying transformations
  - Bits to bytes
  - Bytes to Characters (UTF-8 encoding)
  - Removal of blank lines
  - Removal of C-style comments

Layered Translations Matrix w/Dynamic Size Example

- The data now looks like:

```
3  2
1  2  3
4  5  6
```

- Let b = blank, n=newline. The data really is this string of characters:

3b2n1b2b3bbn4b5bbb6n

References:
Matrix w/Dynamic Size Example

- DFDL wants to make invalid mistakes like:

```
3 2
1 2
3 4 5 6
```

(line structure doesn't match dimensions) or:

```
3 2
1 2 3
4 5 6
7 8 9
```

(too many rows)

References
Matrix w/Dynamic Size Example

```
<element name="example2">
<sequence>
  <element name="dims">
    <sequence>
      <annotation><appinfo>
        <dfdl:terminator value="\p{whitespace}+\p{Line_Separator}"/>
        <dfdl:separator value="\p{whitespace}"/>
      </appinfo></annotation>
      <element name="xdim" type="int"/>
      <element name="ydim" type="int"/>
    </sequence>
  </element>
<element name="ydata" minOccurs=0 maxOccurs="unbounded">
    <annotation><appinfo>
      <dfdl:separator value="\p{whitespace}+\p{Line_Separator}"/>
    </appinfo></annotation>
    <sequence>
      <element name="xdata" type="double"
               minOccurs="0" maxOccurs="unbounded">
        <annotation><appinfo>
          <dfdl:separator value="\p{whitespace}"/>
        </appinfo></annotation>
      </element>
    </sequence>
  </element>
```

References
Matrix w/Dynamic Size Example

```
<element name="example2">
<sequence>
  <element name="dims">
    <sequence>
      <annotation> … </annotation>
      <element name="xdim" type="int"/>
      <element name="ydim" type="int"/>
    </sequence>
  </element>
<element name=“ydata“ minOccurs=0 maxOccurs=“unbounded”>
    <annotation><appinfo>
      <dfdl:separator value="\p{whitespace}+\p{Line_Separator}"/>

      <dfdl:validation expr="{ $./$arrayLength = $../dims/ydim }"/>

    </appinfo></annotation>
    <sequence>
      <element name=“xdata” type=“double”
               minOccurs=“0” maxOccurs=“unbounded”>
        <annotation><appinfo>
          <dfdl:separator value="\p{whitespace}"/>

          <dfdl:validation expr="{ $./$arrayLength = $../dims/xdim }"/>

        </appinfo></annotation>
      </element>
```

Layered Translations Matrix w/Dynamic Size Example

- Now add in the layered transformations of the streams….

```
<annotation><appinfo>

<container name="charStream" type="string">
  <rep charset="UTF-8"
       container="byteStream"> <!-- a built in container -->
    <valueCalc exp="{ bytesToChars() }"/>
  </rep>
</container>

<container name="noCommentsStream" type="string">
  <rep container="charStream">
    <valueCalc exp="{replaceString( '...a regexp for comments...', ' ')}"/>
  </rep>
</container>

<container name="noBlankLinesStream" type="string">
  <rep container="noCommentsStream">
    <valueCalc exp="{ replaceString( '..a regexp for blanklines..',' ')}"/>
  </rep>
</container>

</appinfo></annotation>
```

Modularity

- Consider this example

```
<xs:simpleType name="binaryInt">
     <xs:restriction base="xs:int">
          <xs:annotation><xs:appinfo>
                    <compositeMapping>
                         <mapping name="data-bytes"/>
                         <mapping name="bytes-int"/>
                    </compositeMapping>
               </xs:appinfo></xs:annotation>
     </xs:restriction>
</xs:simpleType>
```

- This connects the definition of binaryInt all the way back to how bits are turned into bytes
- This over-specification limits reusability

Modularity

- Issue: Why should binaryInt care about where the bytes come from?
  - They could come from a binary file
  - They could come from conversion of uuencoded text back into binary data
  - They could come from decompression.
- DFDL defined types want to be parameterized by where they get their underlying data

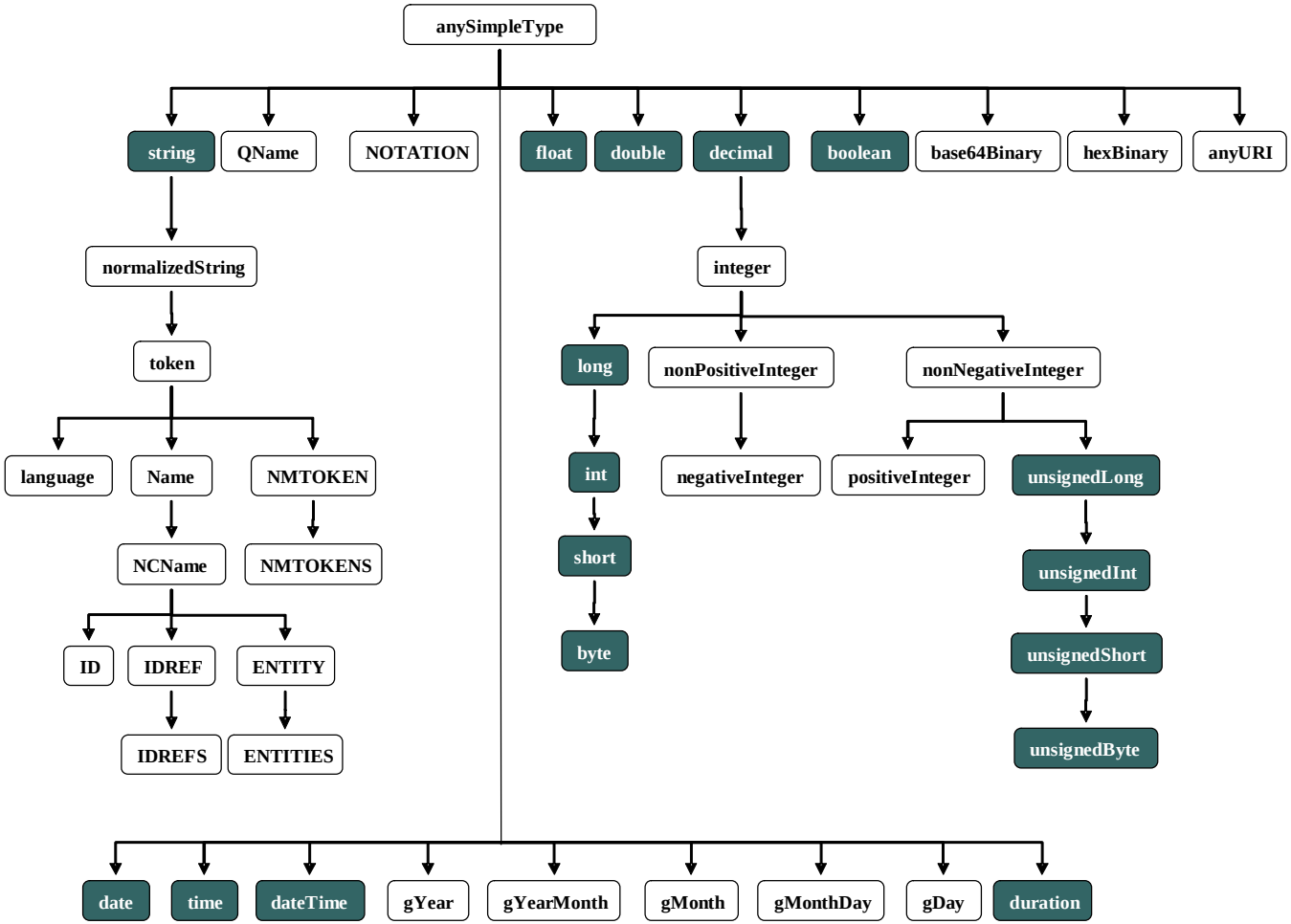
Data Model and Mapping Primitives

- XML/XSD Issues
- Mapping primitives
  - Binary and Text

XML/XSD Issues

- Present type model from XSD
- Multi-dimensional arrays
- Missing types?
  - Basic ones: extended precision float
  - Standardized ones that could be built by users but need to be there: complex numbers?
- Escape sequences needed for XML-illegal char codes. E.g., no &#0; allowed.

XML/XSD – basic types



XSD Complex Types

- Sequence, All
- Choice
- Vectors
  - Any element can have minOccurs and maxOccurs specified.
  - Multi-dimensions only via nested vectors

Multidimensional Arrays

- Nested arrays make storage order explicit
- That is, it's *always* row major order.
  - Last subscript changes first
  - MxN matrix A[i,j] is at (i*N+j)
- What if data is stored column-major order
  - First subscript changes first
  - MxN matrix A[i,j] is at (i+M*j)
- To solve this we need a matrix element type
  - So we can put an arrayStorageOrder property on it!
- Extend XML? Or not?
- No extension proposal:

```
<element name="mymatrix" minOccurs="0" maxOccurs="unbounded"
         type="…the element type…">
   <annotation><appinfo>
     <dimensions storageOrder="rowMajor">
        <dimension lowerLimit="-5" upperLimit="+5"/>
        <dimension lowerLimit="-2" upperLimit="+2"/>
     </dimensions>
   </appinfo></annotation>
</element>
```

Mapping Primitives

- Discussion and Proposals??