Proposed General Purpose Data Containers

Presented at WPEC 38

21 May 2015

Philosophy since last meeting

- Reduce containers to a more basic set
- In part, reduce the number of options available
- This makes programming simpler
- For example, XYs container no longer has mixed and component representations

Classifying containers

- Basics
  - text, values
  - array
- Axis/Axes
  - axis and axes
- Functionals – f(x), f(x,y), f(x,y,z), …
  - XYs, series
  - multiD_XYs, gridded
  - Regions
- Uncertainty
  - uncertainty and uncertainties
- General
  - table

Standard attributes

- Standard attributes for many elements
  - index [Integer >=0] used for sorting
  - label [string] meaning defined by community
  - style [string] meaning defined by community
- Standard function attributes
  - value [see valueType] meaning defined by parent element
  - valueType [one of the supported types]
  - some elements may also have
    - interpolation
    - dimension
- Function elements
  - axes

text xData element

- Parent xData elements: None
- Child elements: None
- Attributes:
  - Common attributes
  - encoding: [String][default=“ascii”]
    - “utf8”, “ascii”, etc.
  - markup: [String][default=“none”]
    - “xml”, “xhtml”, “latex”, etc.
  - length: [Integer32][optional]
- Body: any characters allowed by type and markup attributes.

```
<text markup=“latex”>$\alpha \times x^{3/2}$</text>
```

values xData element

- Parent xData elements: ‘axis’, ‘XYs’, ‘series’, ‘array’?
- Child elements: None
- Attributes:
  - Common attributes
  - valueType: [String][default=“Float64”]
  - sep: [Char][default=“ ”]
  - length: [Integer32][optional] total number of values present
  - start: [Integer32][default=“0”] only values between [start, start+length) are listed, all other values are zero which have been trimmed.
  - size: [Integer32][default=length] total number of values including trimmed zeros. Required if zeros have been trimmed.
- Body: list of ‘type’ entities separated by ‘sep’ and white spaces.

array xData element

- Parent xData elements: ‘gridded’
- Child elements: determined by the compression attribute
- Attributes:
  - Common attributes
  - shape: [List of Integer32s][required]
  - compression: [String][default=“full”] “none”, “diagonal”, “flattened” or “embedded”.
  - triangular: [String][default=‘none’] “upper”, “lower”
    - What about ‘anti’.
  - permutation: [String][default=“+”] one of “+” or “-”
  - storageOrder: [String][default=‘row-major’] “row-major” and “column-major”
  - offset: [List of Integer32s][default=‘0’]
- Body: list of child elements and white spaces

Child elements for each compression type

- compression="none" child elements:
  - a single "values" xData element
- compression="diagonal" child elements:
  - a "values" xData element containing the startingIndices. Data type is an integer and label is "startingIndices"
    - This is optional. Default starting indices are all 0.
  - a "values" xData element containing the values
- compression="flattened" child elements:
  - a "values" xData element with label="flatIndices" [required]. Data type is integer.
  - "values" xData element with label="numberOfValues" [required]. Data type is integer.
  - "values" xData element containing the values [required]
- compression="embedded" child elements:
  - list of "array" xData elements. Each sub-array must have a 'startingIndices' attribute to indicate where it is embedded in the parent array

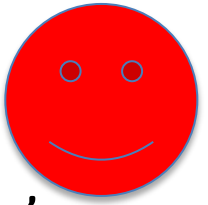
axes xData element

- Parent elements: ‘XYs’, ‘series’, ‘multiD_XYs’-like, ‘regions’-like, ‘gridded’
- Child elements: ‘axis’
  - There must be dimension + 1 axis elements where dimension is the dimension of the parent element.
  - If axes is linked, only axis elements with different values are needed; other axis attributes are derived from link.
- Attributes:
  - Common attributes
- Body: only listed child elements and white spaces are allowed

axis xData element

- Parent element: ‘axes’
- Child elements: ‘values’
  - The ‘values’ element only appears on axis elements inside a ‘gridded’ container
- Attributes:
  - Common attributes
    - ‘index’: [Integer32][required] value is ‘0’ for dependent axis and for the independent axis the subscript value for each x_i in $x_0(x_n, x_{n-1}, ..., x_2, x_1)$
    - ‘label’: [UTF-8][required]
    - ‘style’: [String][optional] If present, a ‘values’ sub-element is required to specify the grid. Values are “points”, “boundaries” or “parameters”
  - ‘unit’: [Unit][default=“”] value is the unit for this axis’ data
  - interpolation: [UTF8Text, contingent]
  - interpolationQualifier: [UTF8Text, contingent]
  - ‘link’: [?][optional]
    - If axis is linked, only ‘axis’ attributes with different values are needed; other values are derived from link. Same for uncertainty sub-element.
- Body: only listed child elements and white spaces are allowed

XYs xData element



- Parent xData elements: ‘regions’, ‘multiD_XYs’, ‘uncertainty’
  - The ‘regions’ and ‘multiD_XYs’ must be of dimension 2.
- Child elements: ‘axes’, ‘values’, ‘uncertainties’
- Attributes:
  - Common attributes
  - Common-2 attributes
  - interpolation: [String][default=“lin,lin”]
- Body: list of child elements and white spaces.
- Restrictions: No discontinuity allowed in values

What about Precision and significant digits?

series xData element

- Parent xData elements: ‘regions’, ‘multiD_XYs’, ‘uncertainties’
  - The ‘regions’ and ‘multiD_XYs’ must be of dimension 2.
- Child elements: ‘axes’, ‘values’, ‘uncertainties’
- Attributes:
  - Common attributes
  - Common-2 attributes
  - function: [String][required]
    - Pre-defined types: “Legendre” and “polynomial”
  - lowerIndex: [Integer32][default=“0”]
  - domainMinimum: [Determined by value’s ‘valueType’][depends on function]
  - domainMaximum: [Determined by value’s ‘valueType’][depends on function]
- Body: list of child elements and white spaces.

multiD_XYs xData element

- Parent xData elements: any multiD_XYs container, ‘regions’, ‘uncertainties’
  - The parent multiD_XYs shall have dimension one higher than self.
  - The parent region shall have the same dimension as self.
- Child elements: ‘axes’, ‘uncertainties’, any (dimension-1)-D functional xData container
- Attributes:
  - Common attributes
  - Common-2 attributes
  - dimension, interpolation, interpolationQualifier
- Body: list of child elements and white spaces.

regions xData element

- Parent xData elements:, ‘uncertainties’, any (dimension+1) functional xData container
- Child elements: ‘axes’, ‘uncertainties’, any (dimension-1) functional xData container
- Attributes:
  - Common attributes
  - Common-2 attributes
  - dimension
- Body: list of child elements and white spaces.

gridded xData element

- Parent xData elements: ‘uncertainties’
- Child elements:
  - ‘axes’ (whose independent ‘axis’ elements each has a ‘grid’)
  - ‘array’
  - ‘uncertainties’
- Attributes:
  - Common attributes
  - Common-2 attributes
- Body: list of child elements and white spaces.

uncertainty xData element

- Parent element: ‘uncertainties’
- Child elements: one functional xData element of the proper dimension
- Attributes:
  - Common attributes
  - ‘relation’: [String][default=“absolute”] shall be “absolute”, “relative” or “percent”.
  - ‘type’: [String][default=“single”] may be “variance”, “variance-”, “variance+”, “covariance”, “confidence-interval”, etc.
  - ‘pdf’: [String][default=“normal”] “normal”, “log-normal”, etc.
- Body: only listed child elements and white spaces are allowed.

uncertainties xData element

- Parent element: any functional xData element
- Child elements: zero or more ‘uncertainty’ elements
- Attributes:
  - Common attributes
- Body: only listed child elements and white spaces are allowed

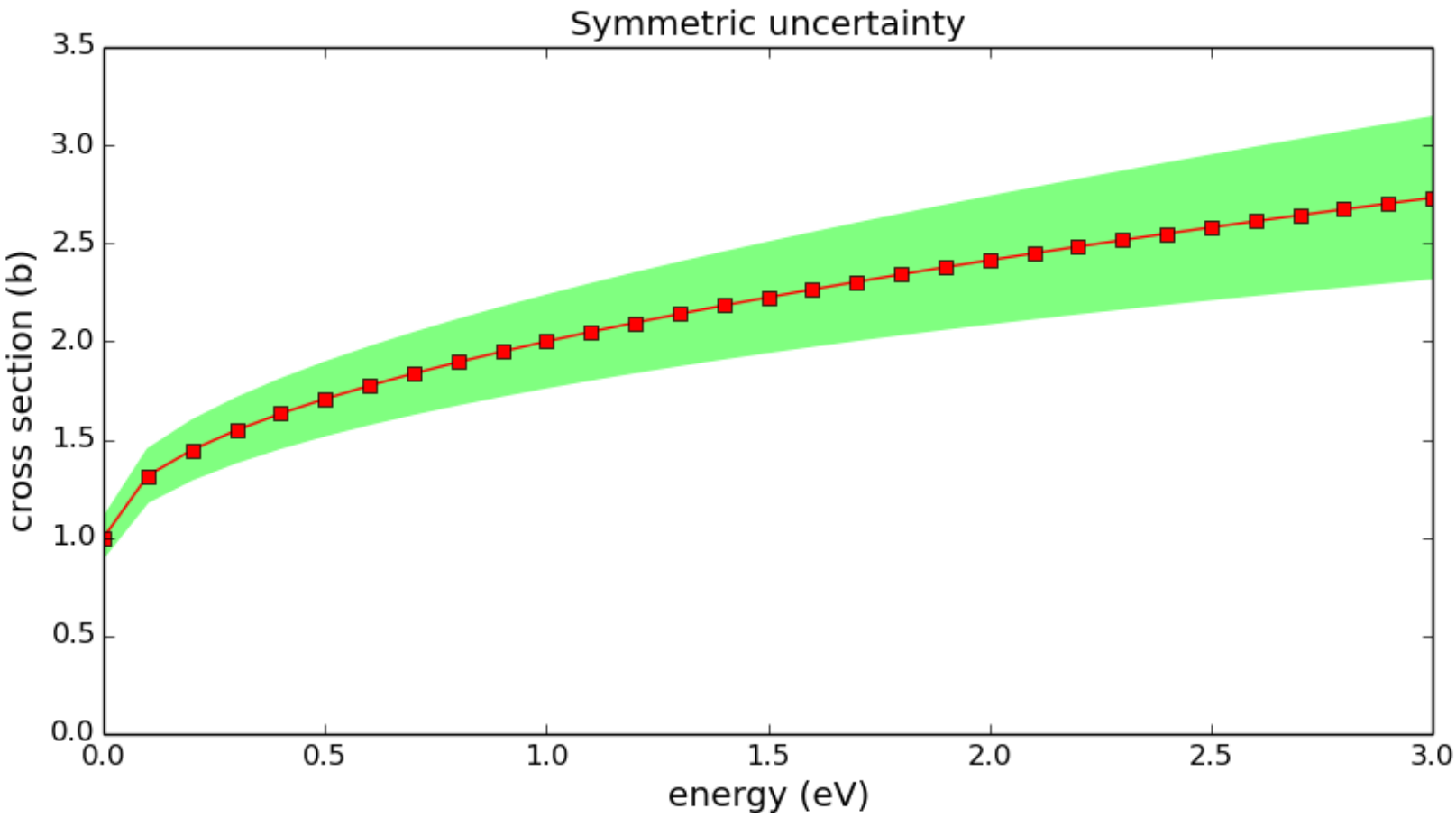
Comment on uncertainties/uncertainty containers

- Allows a cross section's variance/covariance data to reside inside the cross section's container
  - That is, not separated as they are in ENDF

- For regions and multi-dimensional containers, variance/covariance can be at various levels
  - 2d container
    - 1d container with uncertainty data
    - 1d container with uncertainty data
  - 2d container
    - 1d container
    - 1d container
    - uncertainty data – 2d uncertainty

Uncertainty example 1

```
<uncertainties>
    <uncertainty relation="relative”>
        <xys>
            <values> 0.0 0.1 3.0 0.15</values></XYs></uncertainty></uncertainties>
```



Uncertainty example 2

```
<uncertainties>
   <uncertainty type="variance+" pdf="log-normal”>
      <XYs><values> 0.1 0.1 0.4 0.2 1 0.25 1.5 0.25 2.1 0.1</values>
         </XYs></uncertainty>
   <uncertainty type="variance-" pdf="normal”>
      <XYs><values> 0.1 0.6 0.4 0.4 2.1 0.1</values>
         </XYs></uncertainty> </uncertainties>
```

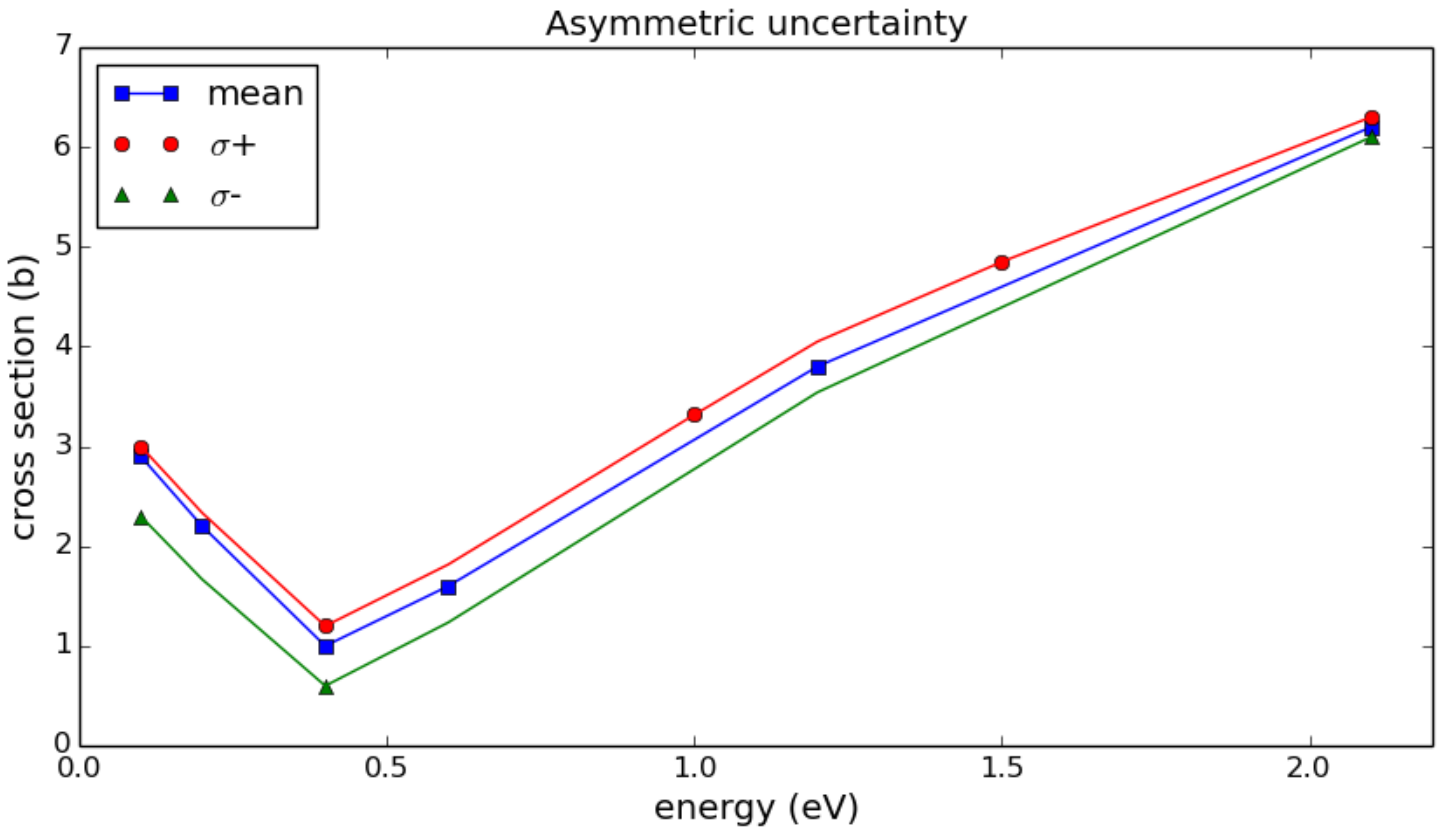


table xData element

- Parent xData elements: None
- Child elements:
- Still TBD
  - Caleb and I are working to simplify this from our previous specifications
  - Ignore it in the documentations

Extras

- style: a simple label to designate different categories of data: For example:
  - temperature
  - different groups
- Links

Style attribute

```
<styles>
  <evaluated name="eval" temperature="0 K" date="2005-12-25" .../>
  <linearize name="L1" source="eval" ...>
  <heat name="T1" ...>   <grouped name="G100F1"...> ... </styles>
```

```
<crossSection>
    <regions dimension="1" style="eval">
     <axes>
      <axis index="1" label="energy_in" unit="eV"/>
      <axis index="0" label="crossSection" unit="b"/></axes>
     <XYs index="0">
      <values length="2750">1e-5 3.842443 1.0931e-5 3.842759 … </XYs>
     <XYs index="0"> …<XYs></regions>
  <XYs style="L1">…</XYs>
  <XYs style="T1">…</XYs>
  <values style="G100F1"> ...</values>
  <values style="G100F2"> ...</values>
  <values style="G1000F1"> ...</values>
  <values style="G1000F2"> ...</values></crossSection>
```