

SBGN-ML

Milestone 1

Generalities

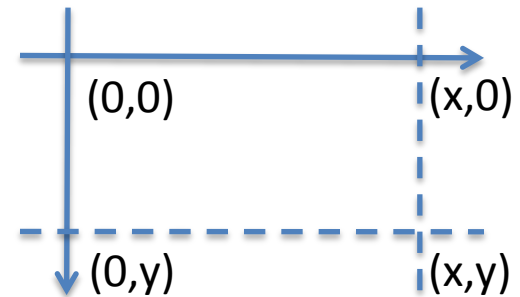
- SBGN-ML is XML based
 - cf. schema SBGN.xsd
- SBGN-ML files represent SBGN maps
 - PD (Process Description) only in milestone 1

Basic geometry and labels

COMMON TYPES AND ELEMENTS

Point

- **PointAttributes** attribute group
 - absolute 2D Cartesian coordinates
 - **x** (float) : horizontal, from left to right
 - **y** (float) : vertical, from top to bottom
 - origin in top-left corner of map
 - no unit (scale free)



- **point** element: has **PointAttributes**
 - `<point x="25.4" y="12.3" />`

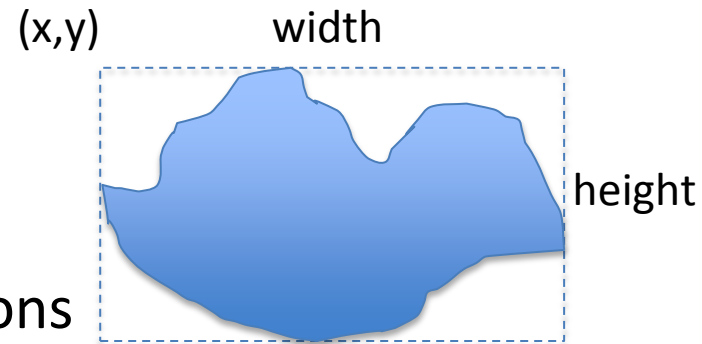
Bounding box

- **bbox** element

- describes a rectangle

- **PointAttributes** = top left corner

- **width & height** (float) = dimensions



- `<bbox x="23" y="11" width="75" height="43">`

- outer limit of a shape (can be irregular)

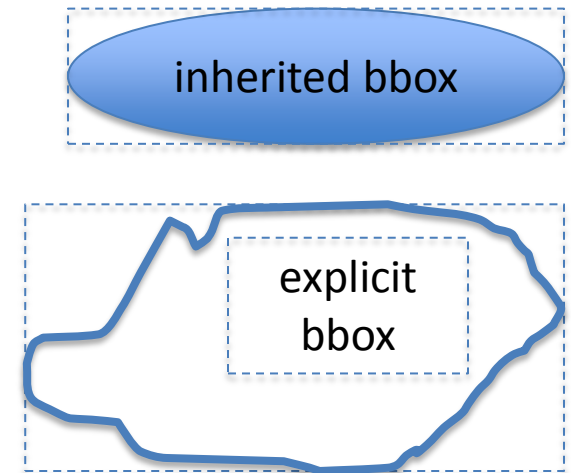
- applies to **glyph** and **label** elements

- for process nodes = central shape only
(not input/output ports)



Label

- **label** element = text of a **glyph** element
 - **text** attribute (string, mandatory)
 - can be multiline: use line break `
`
 - **bbox** child (optional) = position
 - the **text** is centered in the **bbox**
 - use parent **glyph**'s **bbox** when omitted
 - represents **text** size hint when explicit
 - `<label text="inherited bbox">`
 - `<label text="explicit
 bbox">`
`<bbox x="34" y="9" width="60" height="10">`
`</label>`



General structure of a SBGN-ML file

HIGH LEVEL ELEMENTS

Structure of a SBGN-ML file

- **sbgn** root element
 - single **map** element: SBGN PD map
 - list of **glyph** elements (can be empty)
 - list of **arc** elements (can be empty)

- `<sbgn>`
 - `<map>`
 - `<glyph ...>...</glyph>`
 - `<glyph ...>...</glyph>`
 - ...
 - `<arc ...>...</arc>`
 - `<arc ...>...</arc>`
 - ...
 - `</map>`
- `</sbgn>`

Stand-alone high level SBGN nodes, and sub-nodes

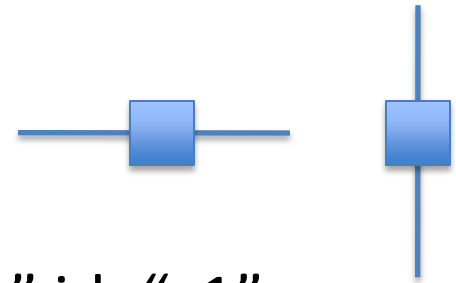
GLYPH ELEMENT

class attribute (optional)

- Semantic of the **glyph**:
 - Shape visual appearance
 - Map syntactic validity
- Entity Pool Nodes (EPN):
 - ***unspecified entity*** [DEFAULT]
 - *simple chemical*
 - *macromolecule*
 - *nucleic acid feature*
 - *complex*
 - *simple chemical multimer*
 - *macromolecule multimer*
 - *nucleic acid feature multimer*
 - *complex multimer*
 - *source and sink*
 - *perturbing agent*
- Process Nodes (PN):
 - *process*
 - *omitted process*
 - *uncertain process*
 - *association*
 - *dissociation*
 - *phenotype*
- Logic Operator Nodes (LON):
and, or, not
- Sub-glyphs on Nodes: *state variable, unit of information*
- Sub-glyphs on Arcs: *stoichiometry*
- Other glyphs: *compartment, submap, tag*

Other attributes

- **id** attribute (mandatory)
 - alphanumeric identifier, starting with a letter
 - usually meaningless:
`<glyph id="glyph1234" ...> ... </glyph>`
- **orientation** attribute (optional)
 - Process node: horizontal, vertical
 - Tag: left, right, up, down
 - `<glyph class="tag" orientation="right" id="g1">`
...
`</glyph>`



Text inside a glyph: **label** or **state**

- **state** (optional): state variables only

- **value** attribute

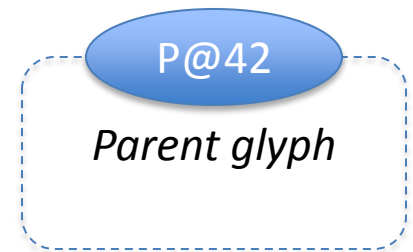
- **variable** attribute

- `<glyph class="state variable" id="g1">`

- `<state value="P" variable="42" />`

- ...

- `</glyph>`



- **label** (optional): all others, except PN and LON

- `<glyph ...> <label text="my label" /> ... </glyph>`

clone element

- Optional
- Means the glyph has a clone marker
 - `<glyph class="simple chemical" id="g1">`
... `<clone />` ...
`</glyph>`
- Can contain a label
 - `<glyph class="complex" id="g1">`
... `<clone>` `<label text="clone label" />` `</clone>` ...
`</glyph>`



Geometry of a glyph: **bbox**

- Mandatory
- `<glyph id="g1">`
 `<bbox x="2" y="5" width="67" height="24" />`
 ...
 `</glyph>`
- Cf. definition of **bbox**

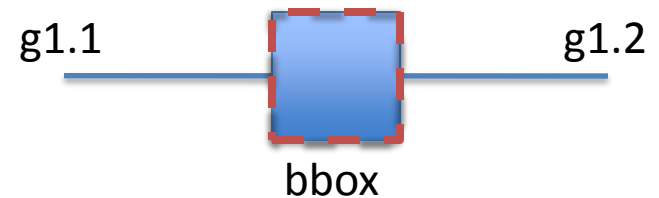
Children elements: **glyph** list

- Optional, any size
- `<glyph id="g1">`
 - ...
 - `<glyph id="g1.1"> ... </glyph>`
 - `<glyph id="g1.2"> ... </glyph>`
 - ...
 - `</glyph>`
- Examples:
 - Unit of information of a compartment
 - State variable of an EPN
 - Inner EPN-like component of a complex

Children elements: **port** list

- Optional, any size
 - 2 **port** elements required for PN (in & out ends)
- Attributes:
 - **PointAttributes** absolute 2D Cartesian coordinates
 - **id**: alphanumeric identifier starting with a letter
- `<glyph class="process node" id="g1" orientation="horizontal">`

```
...  
<port x="23" y="4" id="g1.1" />  
<port x="56" y="4" id="g1.2" />  
...  
</glyph>
```



Visible link between two nodes

ARC ELEMENT

class attribute

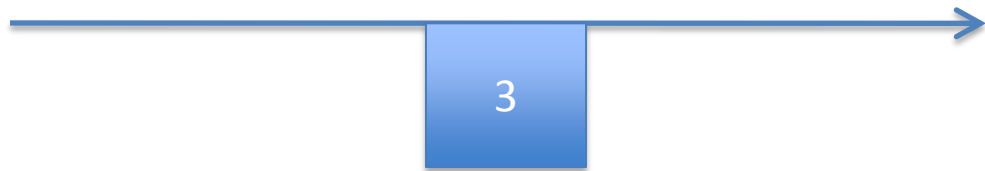
- Semantic of the **arc**:
 - Line visual appearance (usually, decorators at the end)
 - Map syntactic validity (cf. **class** of **source** and **target**)
- *production* and *consumption* arcs
- all types of modification arcs:
 - *modulation*
 - *stimulation*
 - *catalysis*
 - *inhibition*
 - *necessary stimulation*
- *logic arc*
- *equivalence arc*

source and target attributes

- Mandatory
- Can refer to the **id** of a **glyph** or a **port**
- `<arc class="consumption" source="epn1" target="pn1.1">`
...
`</arc>`

stoichiometry of an arc

- Optional child element
- **glyph** of class **stoichiometry**
- `<arc class="production" source="pn1.2" target="epn2">`
 `<glyph class="stoichiometry" id="s1">`
 `<label text="3"/>`
 `<bbox x="23" y="45" width="10" height="10">`
 `</glyph>`
 ...
`</arc>`



Geometric path of an arc

- **start** element
 - **PointAttributes**: start point of the path
- optional list of **next** elements
 - **PointAttributes**: next point in the path
 - **point** elements (between 0 and 2)
 - control points of the Bezier curve (1: quadratic, 2: cubic)
- **end** element
 - **PointAttributes**: end point of the path
 - **point** elements (between 0 and 2)
 - control points of the curve (1: quadratic, 2: cubic)

Example of paths (straight lines)

- Single straight line

```
<arc ...>
```

```
<start x="10" y="20"/>
```

```
<end x="60" y="20"/>
```

```
</arc>
```



- Polyline

```
<arc ...>
```

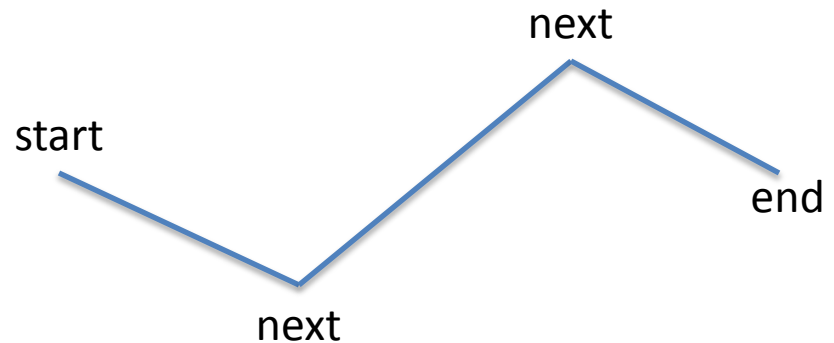
```
<start x="10" y="20"/>
```

```
<next x="30" y="30"/>
```

```
<next x="50" y="10"/>
```

```
<end x="60" y="20"/>
```

```
</arc>
```



Example of paths (curves)

- Quadratic Bezier curve

```
<arc ...>  
  <start x="10" y="20" />  
  <end x="60" y="20">  
    <point x="34" y="23" />  
  </end>  
</arc>
```
- Cubic Bezier curve

```
<arc ...>  
  <start x="10" y="20" />  
  <end x="60" y="20">  
    <point x="12" y="23" />  
    <point x="34" y="21" />  
  </end>  
</arc>
```

Example of paths (mixed)

- `<arc ...>`
 - `<start x="10" y="20" />`
 - `<next x="30" y="30" />`
 - `<next x="50" y="10">`
 - `<point x="14" y="53">`
 - `</next>`
 - `<end x="60" y="20">`
 - `<point x="12" y="23" />`
 - `<point x="34" y="21" />`
 - `</end>`
- `</arc>`