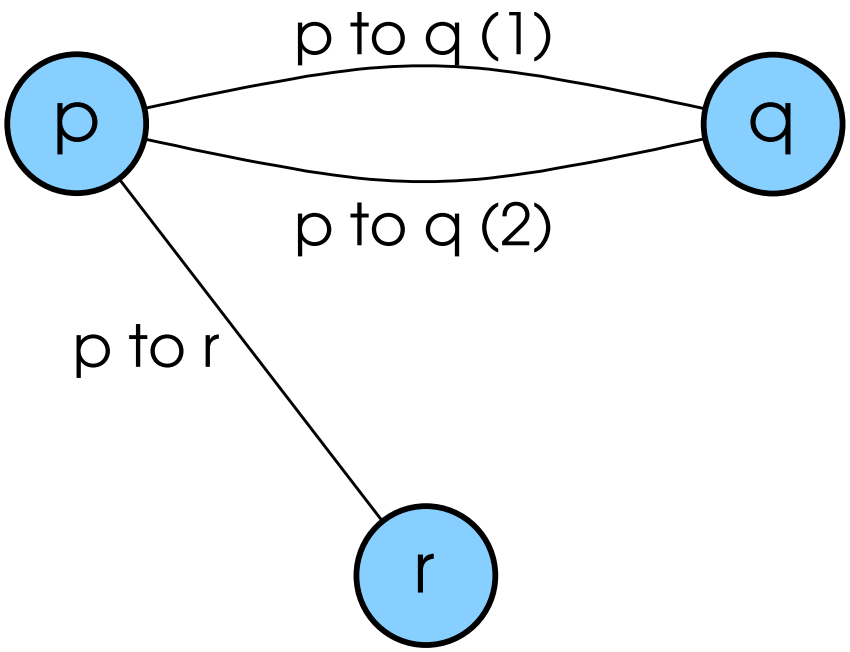
Using GXL and SVG for Describing Graphs with Layout

Mark Minas
minas@acm.org

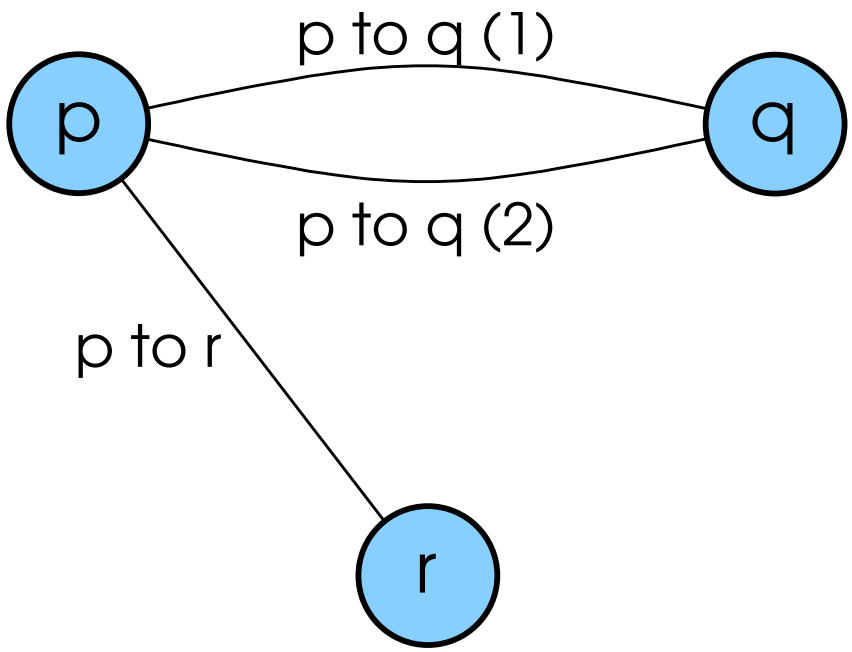
Problem

Given a graph represented by GXL:

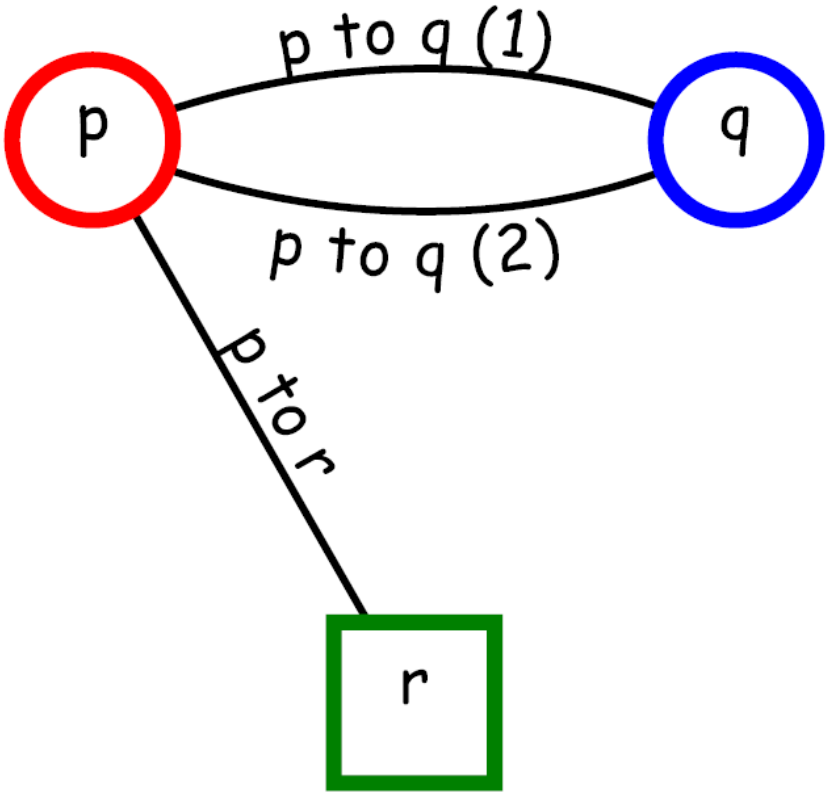


Problem

Given a graph represented by GXL:



How do we describe its layout like the following one ?



Layouted graph

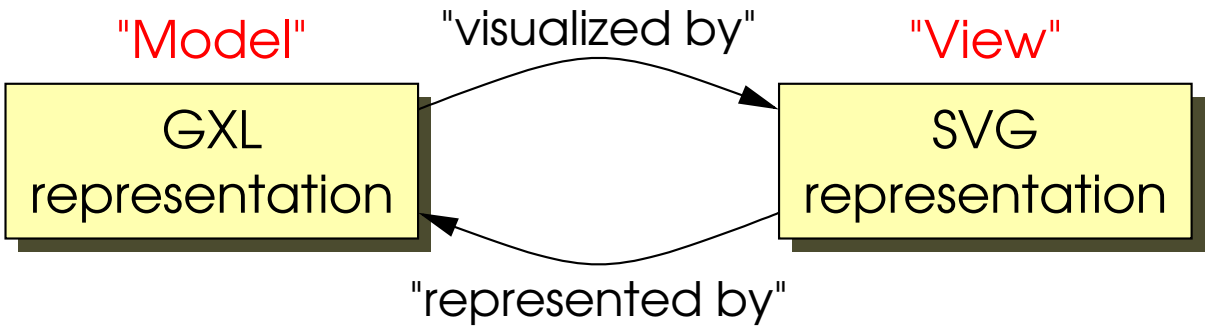
GXL representation of the graph (cont.)

```
  <!-- edge p to q (1) -->
  <edge id="a2" from="n2" to="n3">
    <attr id="a2name" name="name">
      <string>p to q (1)</string>
    </attr>
  </edge>

  <!-- edge p to q (2) -->
  <edge id="a3" from="n2" to="n3">
    <attr id="a3name" name="name">
      <string>p to q (2)</string>
    </attr>
  </edge>
</graph>
</gxl>
```

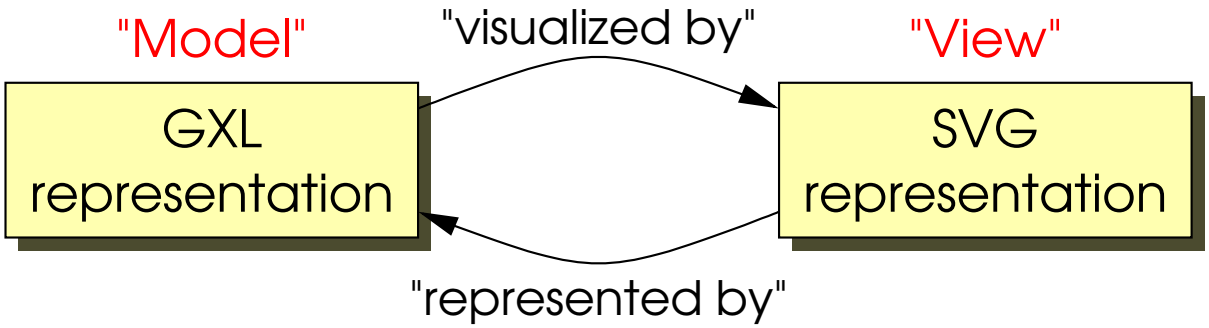
SVG representation of the graph layout

```
<?xml version="1.0" standalone="yes"?>

<svg width="10cm" height="10cm" viewBox="0 0 10 10"
     xmlns="http://www.w3.org/2000/svg"
     xmlns:xlink="http://www.w3.org/1999/xlink">

  <!-- edge p to r -->
  <path id="a1" fill ="none" stroke="black" stroke-width=".05"
        d="M 2 1 L 4 4.5"/>
  <text id="a1name" dy="-.1" text-anchor="middle"
        font-family="Comic" font-size=".4" fill="black">
    <textPath startOffset="50%"
              xlink:href="#a1">p to r</textPath>
  </text>
```

Linking GXL and SVG

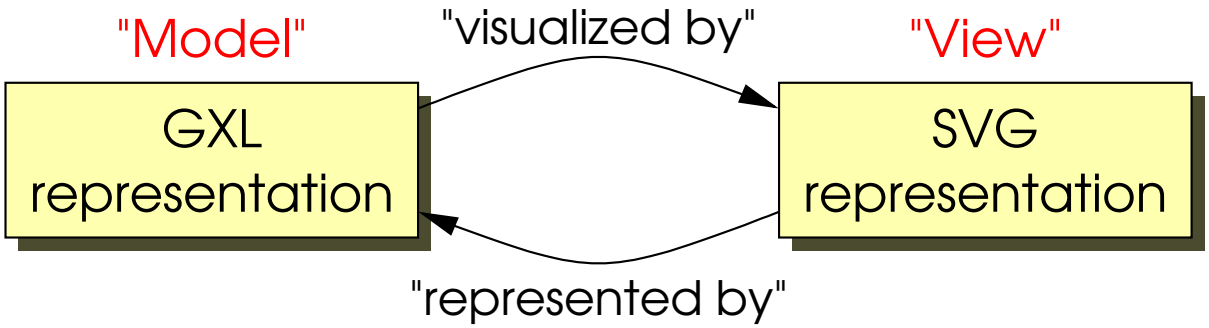


Linking GXL and SVG



Scalable Vector Graphics (SVG) 1.0 Specification
W3C Recommendation 04 September 2001
Current version: `http://www.w3.org/TR/2001/REC-SVG-20010904/`
Section **17. Linking**:

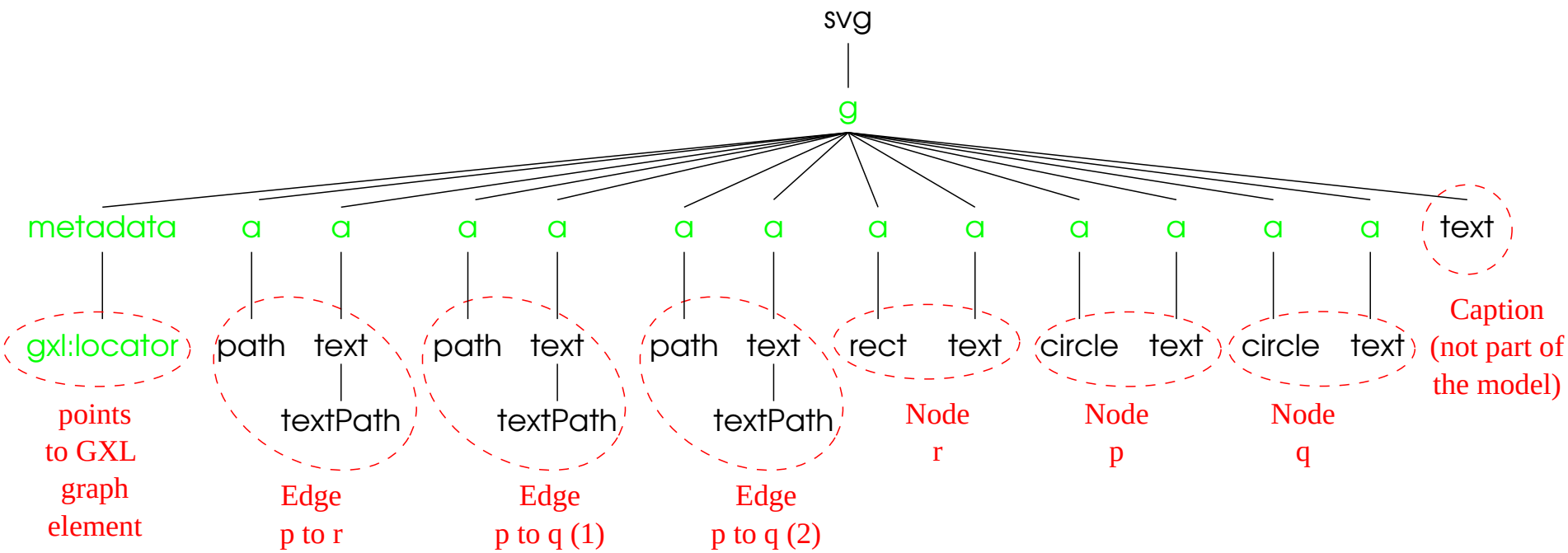
Linking GXL and SVG



Scalable Vector Graphics (SVG) 1.0 Specification
W3C Recommendation 04 September 2001
Current version: `http://www.w3.org/TR/2001/REC-SVG-20010904/`
Section **17. Linking**:

"⟶" SVG provides an a element, analogous to HTML's a element, to indicate links (also known as hyperlinks or Web links). SVG uses XLink for all link definitions.

Structure of the SVG representation with links



SVG representation with links (cont.)

```
<!-- edge p to r -->
<a xlink:href="graphmodel.gxl#a1">
  <path id="a1"
        fill ="none" stroke="black" stroke-width=".05"
        d="M 2 1 L 4 4.5"/>
</a>
<a xlink:href="graphmodel.gxl#a1name">
  <text id="a1name"
        dy="-.1" text-anchor="middle"
        font-family="Comic" font-size=".4" fill="black">
    <textPath startOffset="50%"
              xlink:href="#a1">p to r</textPath>
  </text>
</a>
```

Discussion

- Pro
  - Highly sophisticated graph layout and visual representation using out-of-the-box technology (SVG)
  - Each node, attribute etc. can be visualized differently (cf. red circle vs. green rectangle)

Discussion

- Pro
  - Highly sophisticated graph layout and visual representation using out-of-the-box technology (SVG)
  - Each node, attribute etc. can be visualized differently (cf. red circle vs. green rectangle)
- Contra
  - Verbose

Discussion

- Pro
  - Highly sophisticated graph layout and visual representation using out-of-the-box technology (SVG)
  - Each node, attribute etc. can be visualized differently (cf. red circle vs. green rectangle)
- Contra
  - Verbose
  - Attribute texts in SVG representation are not required to be the same as in the GXL representation.

Discussion

- Pro
  - Highly sophisticated graph layout and visual representation using out-of-the-box technology (SVG)
  - Each node, attribute etc. can be visualized differently (cf. red circle vs. green rectangle)
- Contra
  - Verbose
  - Attribute texts in SVG representation are not required to be the same as in the GXL representation.

    However: That's actually a `Pro`: We have to be able to insert linebreaks etc. anyway.

Discussion

- Pro
  - Highly sophisticated graph layout and visual representation using out-of-the-box technology (SVG)
  - Each node, attribute etc. can be visualized differently (cf. red circle vs. green rectangle)
- Contra
  - Verbose
  - Attribute texts in SVG representation are not required to be the same as in the GXL representation.
    However: That's actually a `Pro`: We have to be able to insert linebreaks etc. anyway.
  - Graph model and graph layout are represented in two different files
    $\Rightarrow$ Consistency problems ?

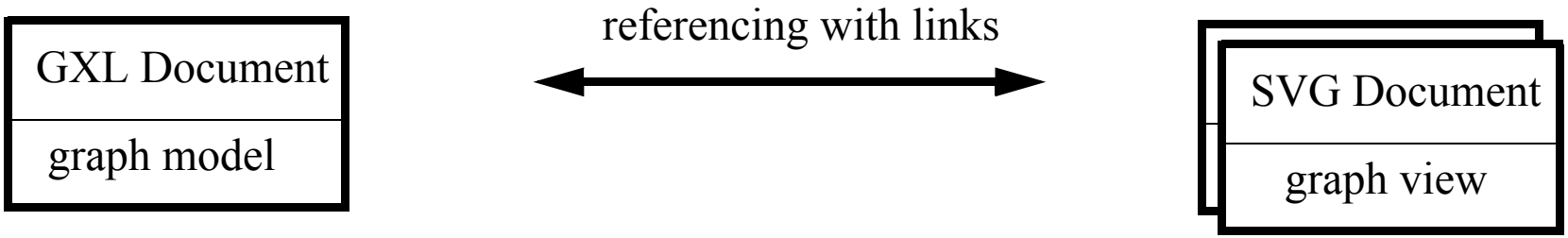
Discussion

- Pro
  - Highly sophisticated graph layout and visual representation using out-of-the-box technology (SVG)
  - Each node, attribute etc. can be visualized differently (cf. red circle vs. green rectangle)
- Contra
  - Verbose
  - Attribute texts in SVG representation are not required to be the same as in the GXL representation.
    However: That's actually a `Pro`: We have to be able to insert linebreaks etc. anyway.
  - Graph model and graph layout are represented in two different files
    $\Rightarrow$ Consistency problems ?
    However: Consistency is a tools problem, not a representation problem!!!

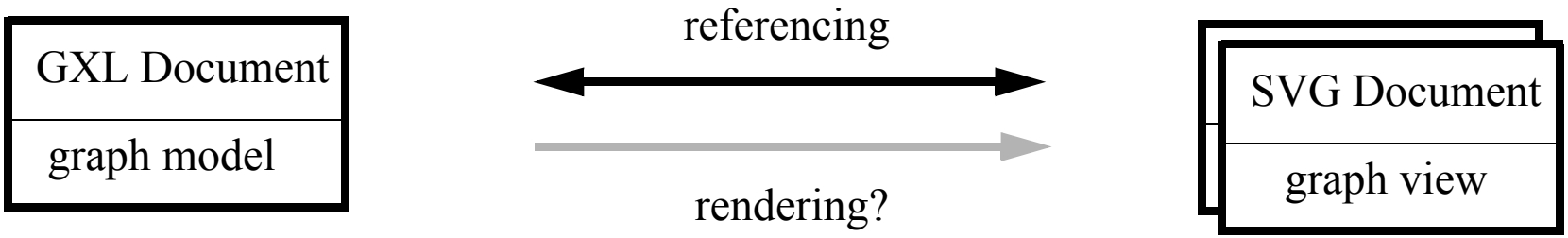
That’s it!

... no, I’m cheating!

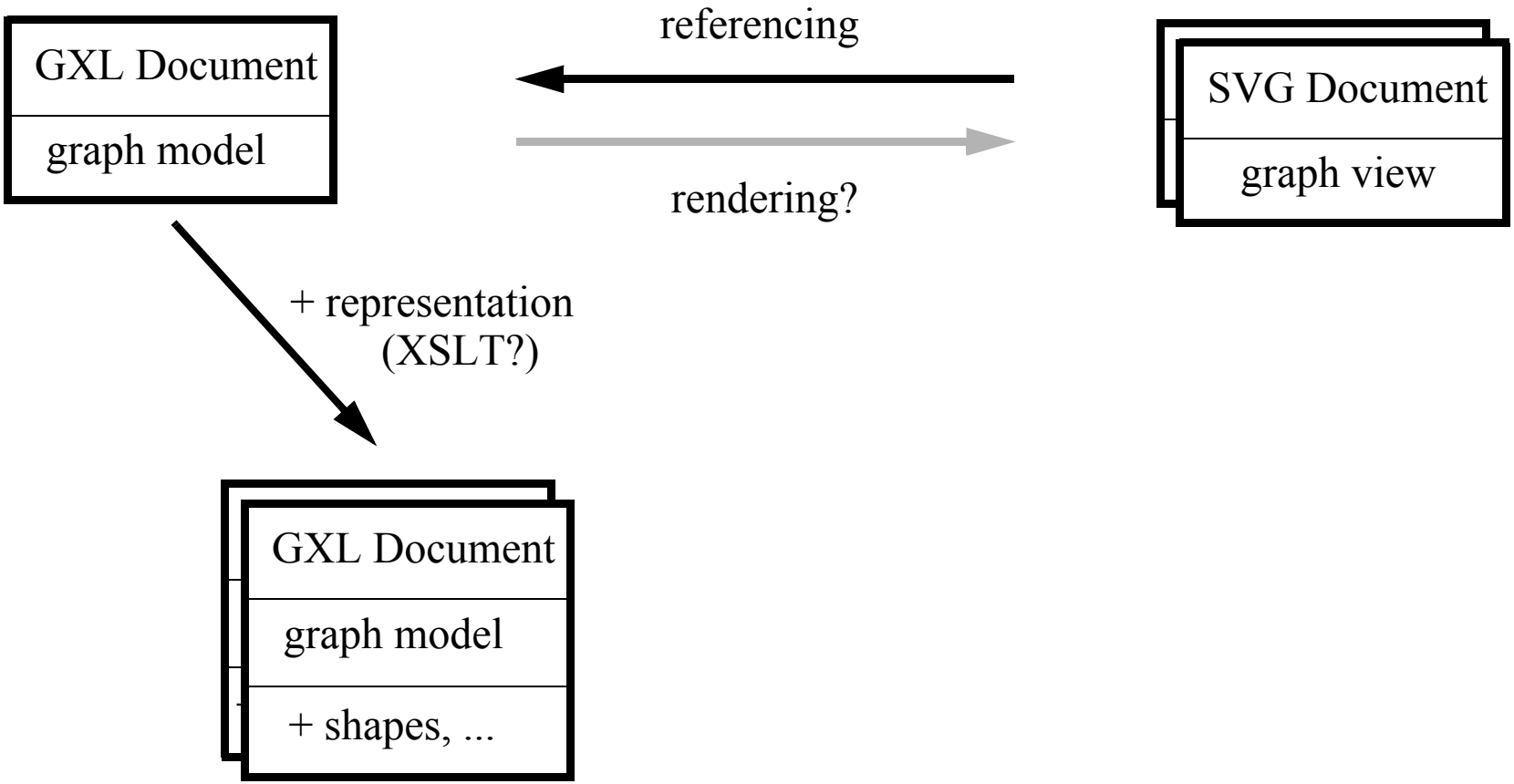
Another Summary of GXL-SVG Idea



Another Summary of GXL-SVG Idea

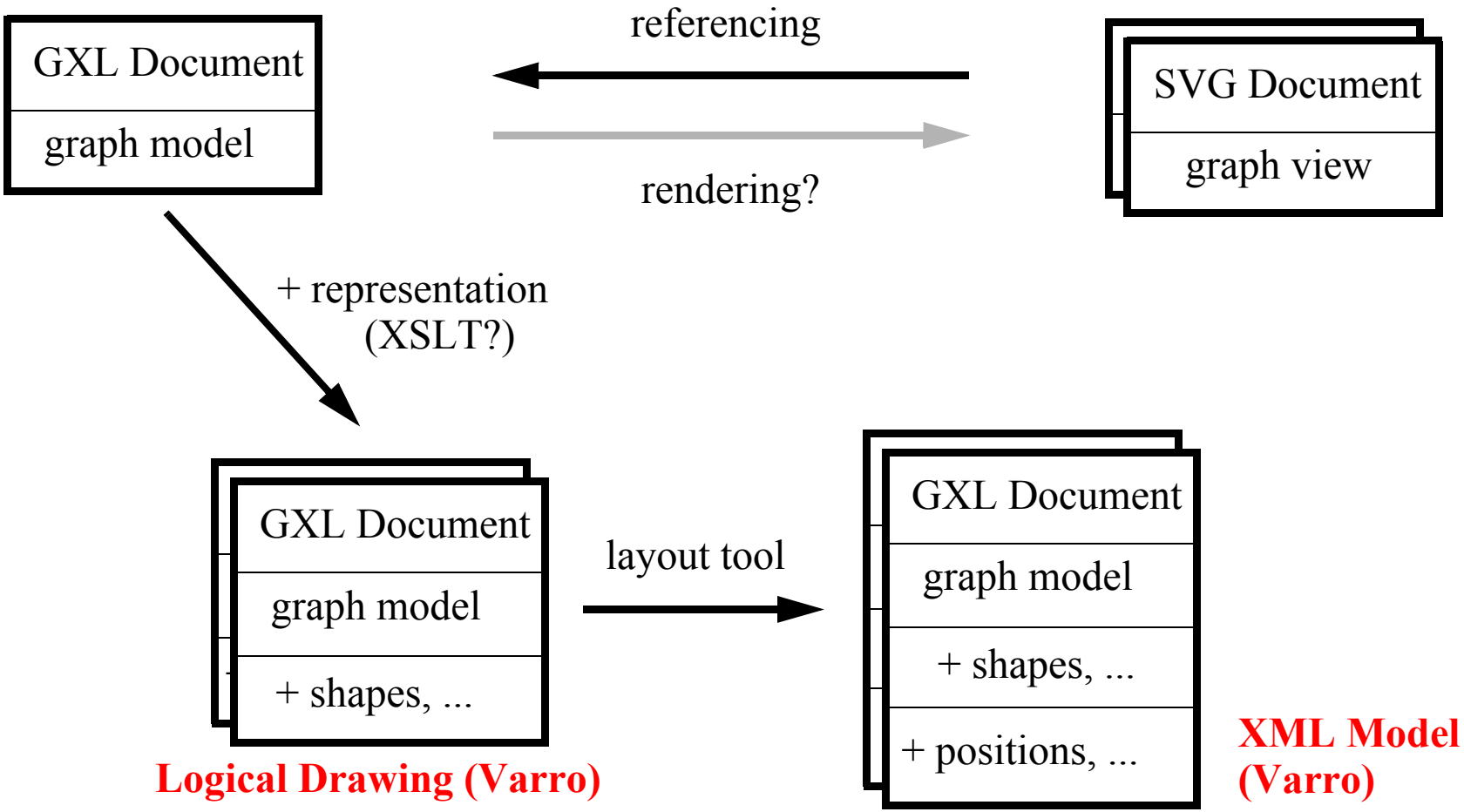


Translating GXL Docs. into SVG Docs.



Logical Drawing (Varro)

Translating GXL Docs. into SVG Docs.



Translating GXL Docs. into SVG Docs.