An Incremental XSLT Transformation Processor for XML Document Manipulation

Lionel Villard – Nabil Layaïda

Presented by Emmanuel Pietriga

Projet Opéra

INRIA Rhône-Alpes

http://www.inrialpes.fr/opera/

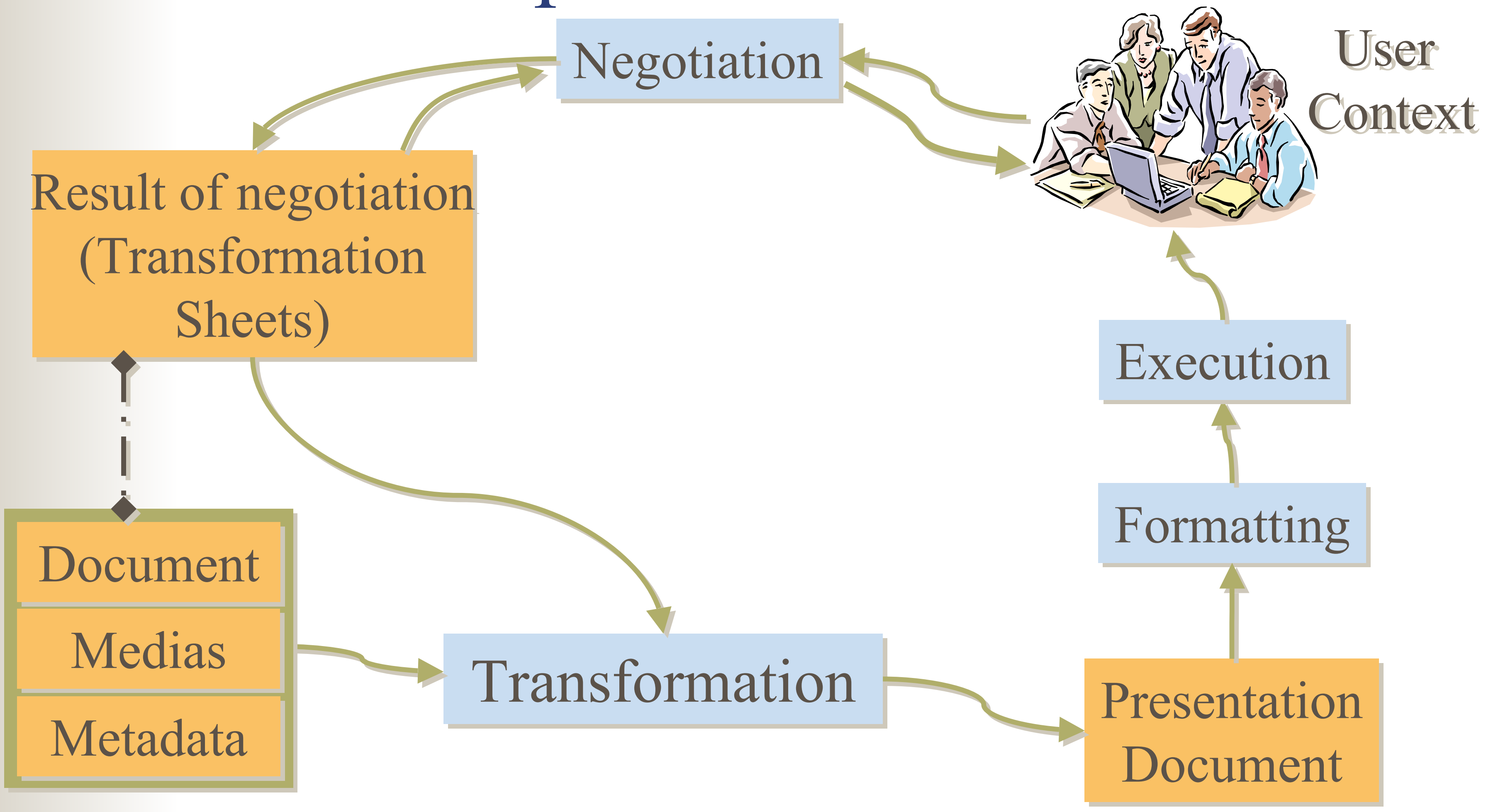
Outline

- Motivation
- Incremental transformations
  - Principles
  - Static Analysis
  - Incremental execution
- Conclusion and perspectives

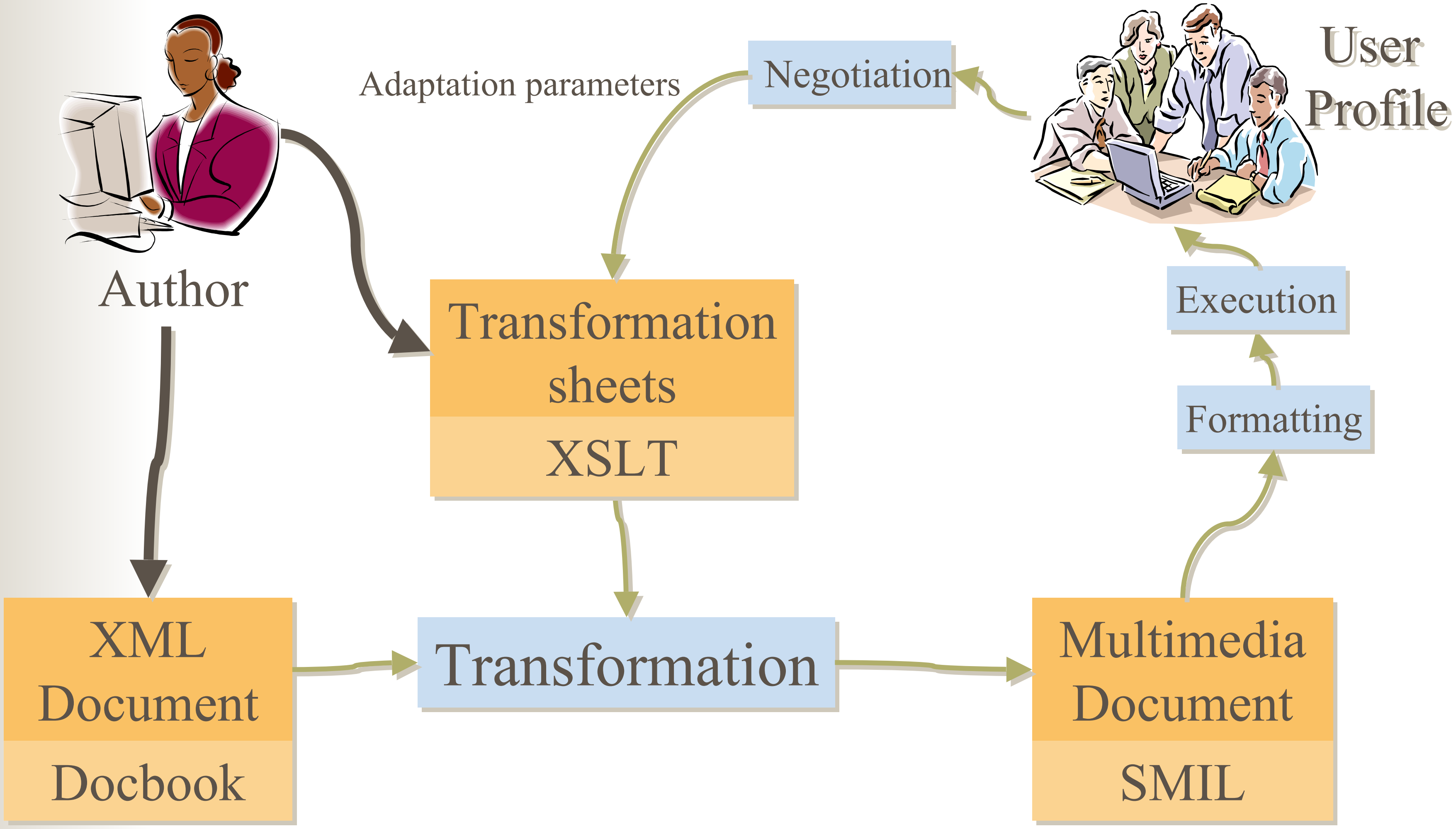
Motivation

- XML Vocabularies
  - Content (logical structure)
  - Presentation (formatting)
- Authoring multimedia presentations for classes of document
- Authoring adaptable presentations

Multimedia presentation architecture



Author involvement



Author skills

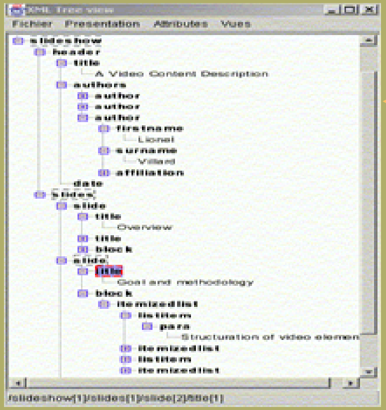
- Classes of document
  - Generally for professionals
  - Need heavy infrastructure (database, schema)
- Adaptable multimedia presentation
  - For any author (novice, professional, etc.)

No transformation coding

Goals

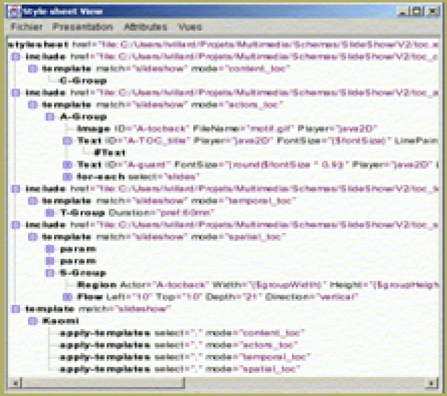
- Author source (content) document and transformation sheet by direct manipulation of target (presentation) document

Source Document (XML)



```
slideshow
  header
    title
      A Video Content Description
    authors
      author
      author
      author
        firstname
          Lionel
        surname
          Villard
        affiliation
    date
  slides
    slide
      title
        Overview
      title
      block
    slide
      title
        Goal and methodology
      block
        itemizedlist
          listitem
            para
              Structuration of video elemen
          itemizedlist
          listitem
          itemizedlist
/slideshow[1]/slides[1]/slide[2]/title[1]
```

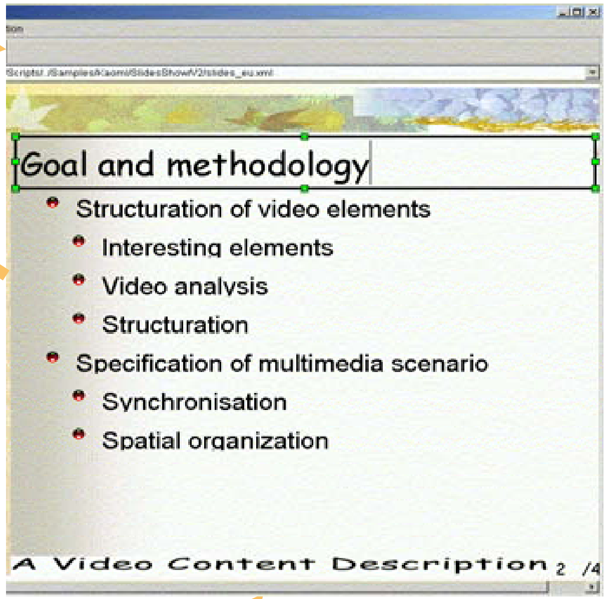
Transformation sheets (XSLT)



```
stylesheet href="file:C:/Users/lvillard/Projets/Multimedia/Schemes/SlideShow/V2/toc_a
  include href="file:C:/Users/lvillard/Projets/Multimedia/Schemes/SlideShow/V2/toc_c
    template match="slideshow" mode="content_toc"
      C-Group
  include href="file:C:/Users/lvillard/Projets/Multimedia/Schemes/SlideShow/V2/toc_a
    template match="slideshow" mode="actors_toc"
      A-Group
        Image ID="A-tocback" FileName="motif.gif" Player="java2D"
        Text ID="A-TOC_title" Player="java2D" FontSize="{$fontSize}" LinePain
          #Text
        Text ID="A-guard" FontSize="{round($fontSize * 0.9)}" Player="java2D" i
        for-each select="slides"
  include href="file:C:/Users/lvillard/Projets/Multimedia/Schemes/SlideShow/V2/toc_s
    template match="slideshow" mode="temporal_toc"
      T-Group Duration="pref 60mn"
  include href="file:C:/Users/lvillard/Projets/Multimedia/Schemes/SlideShow/V2/toc_s
    template match="slideshow" mode="spatial_toc"
      param
      param
      S-Group
        Region Actor="A-tocback" Width="{$groupWidth}" Height="{$groupHeigh
        Flow Left="10" Top="10" Depth="21" Direction="vertical"
  template match="slideshow"
    Kaomi
      apply-templates select="." mode="content_toc"
      apply-templates select="." mode="actors_toc"
      apply-templates select="." mode="temporal_toc"
      apply-templates select="." mode="spatial_toc"
```

2: Authoring

1: Batch Transformation



Scripts/../Samples/Kaomi/SlidesShowV2/slides_eu.xml

Goal and methodology

- Structuration of video elements
  - Interesting elements
  - Video analysis
  - Structuration
- Specification of multimedia scenario
  - Synchronisation
  - Spatial organization

A Video Content Description 2 /4

3: Reverse Transformation
Determine source/transformation modification

4: **Incremental transformation**

Goals

- Author source (content) document and transformation sheet by direct manipulation of target (presentation) document
- Update as fast as possible the target document after modifications of:
  - XML source document(s)
  - Transformation sheet(s)

Incremental transformation processor

- Change only target document fragments that need to be updated w.r.t changes in the source document or in the transformation sheet
- Focus on target document updates due to changes in the source document

XSLT transformations

Source document:

```
<Picture loc="result.jpg" l="100" h="150">
```

Transformation rule:

```
<xsl:template match="Picture">
  <Image src="{@loc}" width="{@l}" height="{@h}>
</xsl:template>
```

Result :

```
<Image src="result.jpg" width="100" height="150">
```

Source modification

Source document:

<Picture loc="result.jpg" l="**300**" h="150">

Transformation rule:

```
<xsl:template match=“Picture">
  <Image src="{@loc}" width="{@l}" height="{@h}>
</xsl:template>
```

Result :

<Image src="result.jpg" width="**300**" height="150">

incXSLT: incremental transformation

Two step process:

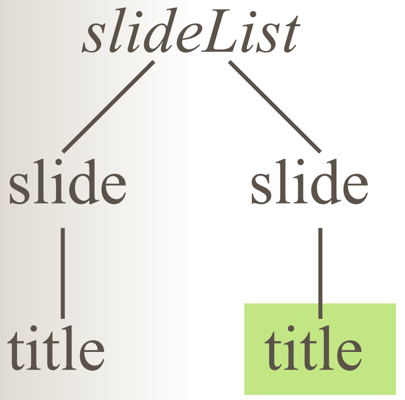
- Preprocessing: static transformation sheet analysis
  - Build template and variable dependency graphs
  - Build re-evaluation rules

    (editing operations, instructions to re-evaluate)
- Incremental processing
  - Run instructions identified thanks to the rules computed during the static analysis

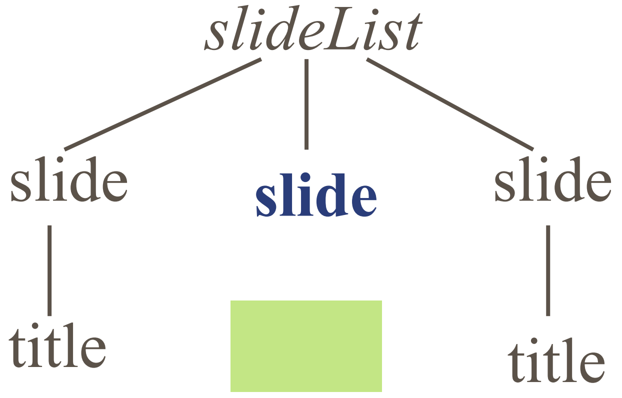
Static analysis example

Expression: slide[position() = 2]/title

Initial evaluation

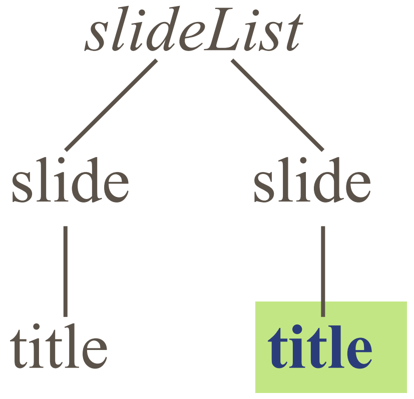


Addition/Removal of slide before the second position



Pattern:
slide[position() <= 2]

Addition/Removal of second slide title

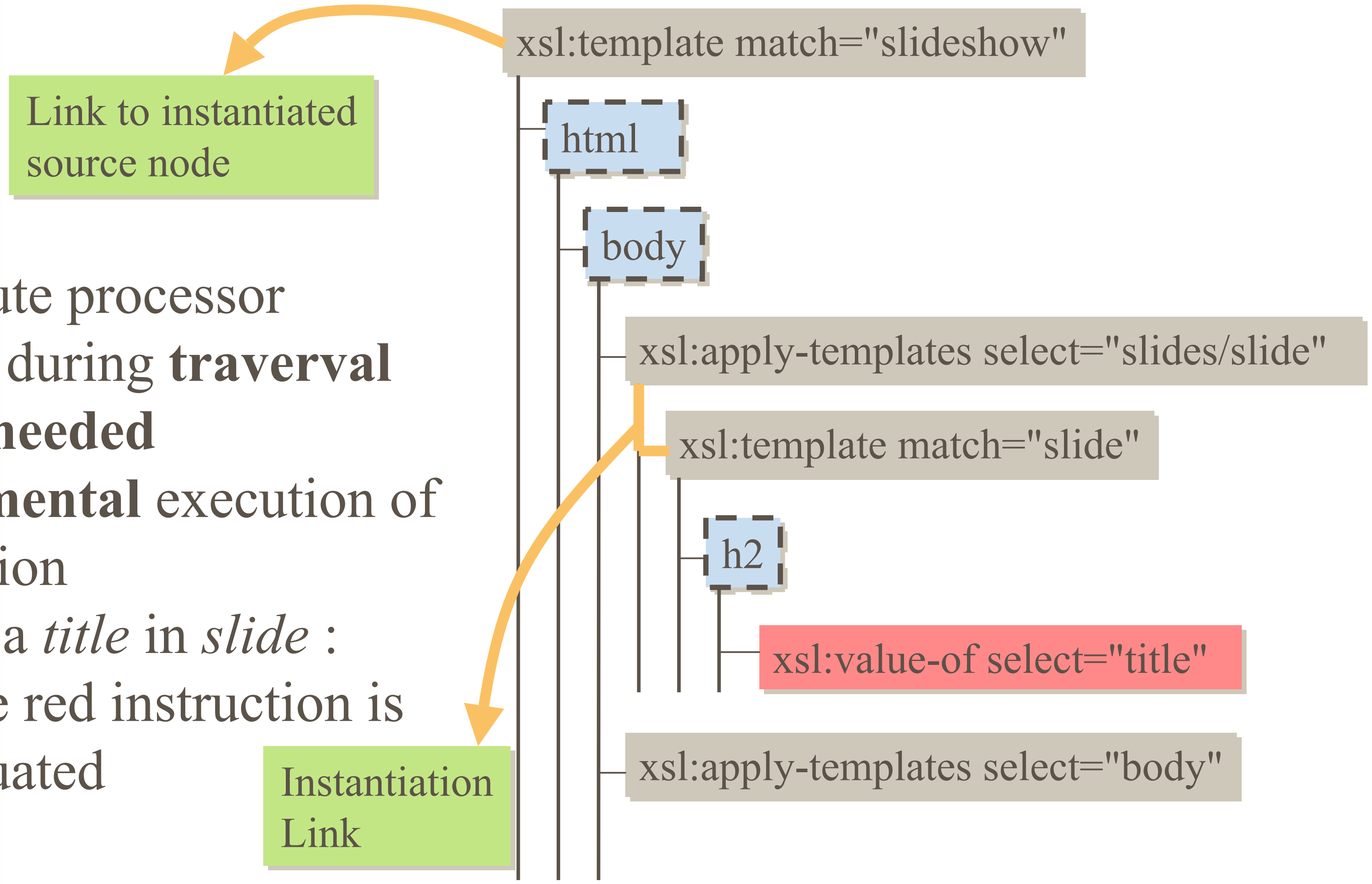


Pattern:
slide[position()=2]/title

Incremental execution

- Set of instructions to re-evaluate, determined during the static analysis
- Restore execution state: execution flow tree
  - Cache data structure
- Two execution models
  - Depth-first execution tree traversal
  - Direct execution of instructions

Execution tree traversal


Link to instantiated source node
xsl:template match="slideshow"
html
body
xsl:apply-templates select="slides/slide"
xsl:template match="slide"
h2
xsl:value-of select="title"
xsl:apply-templates select="body"
Instantiation Link

- Compute processor context during **traverval** and **as needed**
- **Incremental** execution of instruction
- Insert a *title* in *slide* : only the red instruction is re-evaluated

Conclusion

- Experimentation in Xalan (XSLT engine from Apache)
  - Important execution gain
  - Limited memory overhead
- Other applications
  - Batch transformation optimization
  - Web site construction
  - Transformation parameter modifications

Perspectives

- Go further about static analysis
  - Consider schema of source document
- Apply to XPath/XSLT 2.0
  - Propose a formal semantics
  - Enhance static analysis
  - Implications on incremental execution