

Preparing a Critical Edition of an Incunabula with TEI - The Online Edition -

J. Schneeberger

Hochschule Deggendorf

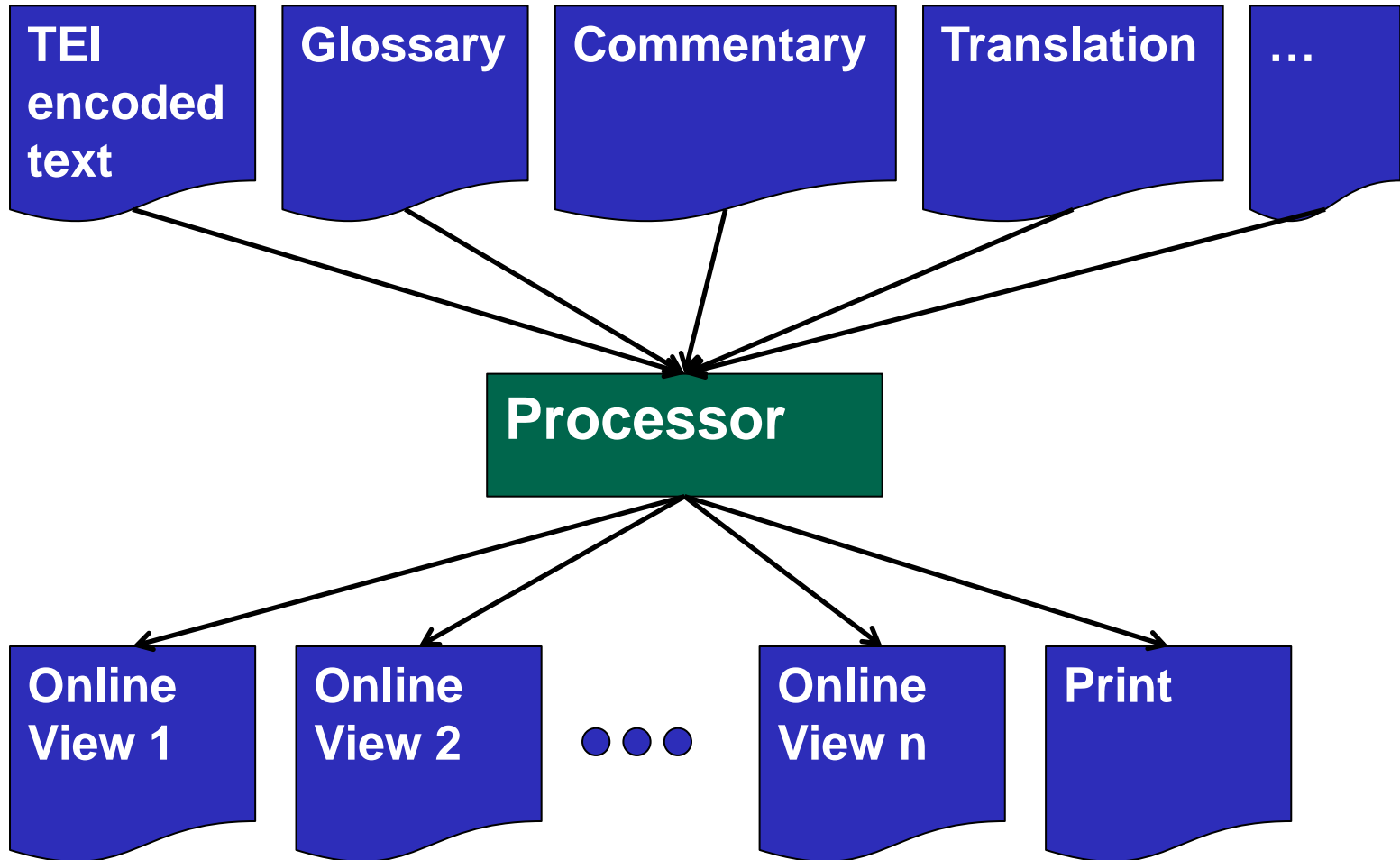
josef.schneeberger@fh-deggendorf.de

Overview

- System Architecture
- XPATH and XSLT
- Tools
 - Eclipse
 - Cocoon
- Information Integration

System Architecture

From Source to View



Sources and Views

- Multiple XML files
 - TEI encoding
 - Frequently modified
 - New glossary
 - New commentary
 - Translation
 - ...
- Images
- HTML Views
 - Side-by-side presentation
 - multiple variants
 - transcription and original version
 - Selector for pages and variants
- Printable Version

Design considerations

- XML based
- Easy to modify
- Multiple contributors

XPath

Sources and Links

- Anders Møller, Michael I. Schwartzbach
“An Introduction to XML and Web
Technologies”
Addison-Wesley, January 2006
- <http://www.brics.dk/ixwt/>

What is XPath?

- A notation to describe (parts of) XML trees
- Navigate trees
- Extract information from trees

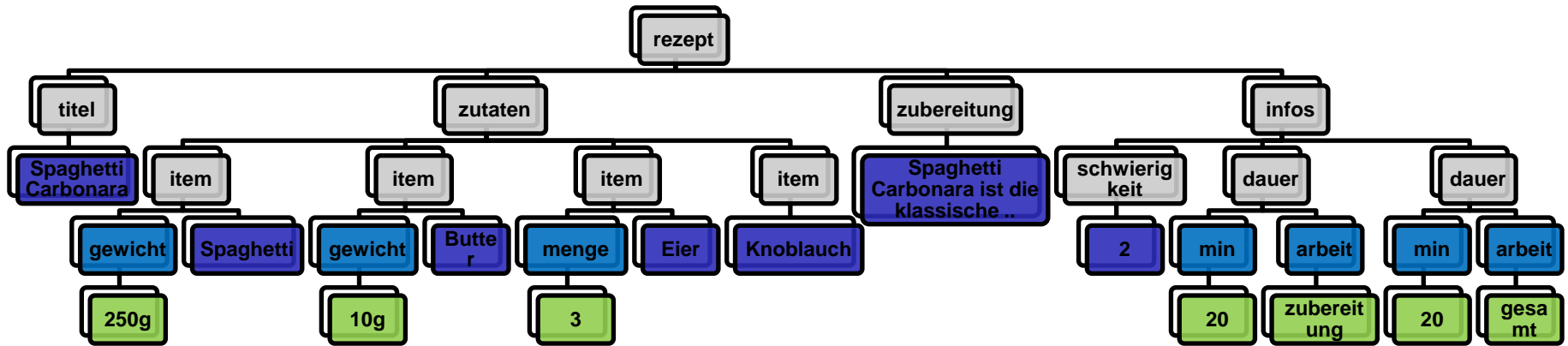
- Used in:
 - XSLT (to transform XML to other formats)
 - XML Schema
 - XLink and XPointer (references)



XPath

- Combines path expressions (known from shell programs) and simple constructs from programming languages
 - *.xml – all files the end with “.xml“
 - E.g.: /body/table[@border="1"]
- An XSLT stylesheet uses XPath expressions as attributes
 - <xsl:template match="/">
 - <xsl:value-of select=".">
 - <xsl:apply-templates select="/recipe/ingredients/item/">

An XML tree (DOM)



Location Path (1)

Two types of location paths

- Relative path
 - One or more steps separated by “/”
 - Starting from a context, each step selects a set of nodes
 - Each selected node is used as a new context
 - The results of the steps are unified
- Absolute path
 - Starting with “/” followed by a relative path.
 - / selects the root of the document.

Location Path (2)

- A location path ...
 - is a sequence of nodes
 - without duplicates.
- General form:
`step1 / step2 / ... / stepN`

Location Step

- A location step consists of:
 - An axis,
 - A node test
 - Expressions – used as predicates.

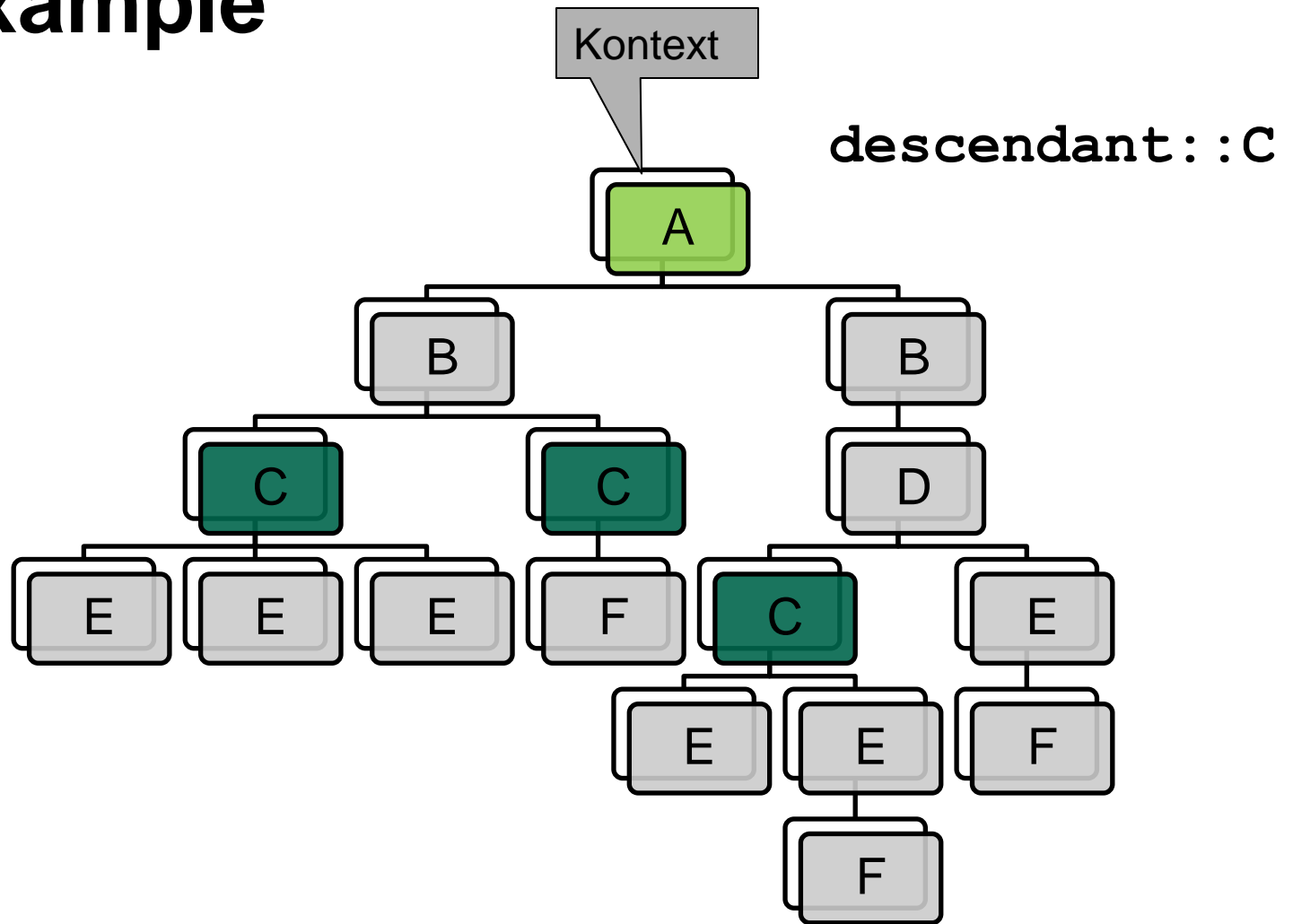
- General form

axis :: nodetest [expr1] [expr2] ...

Evaluation of Location Paths

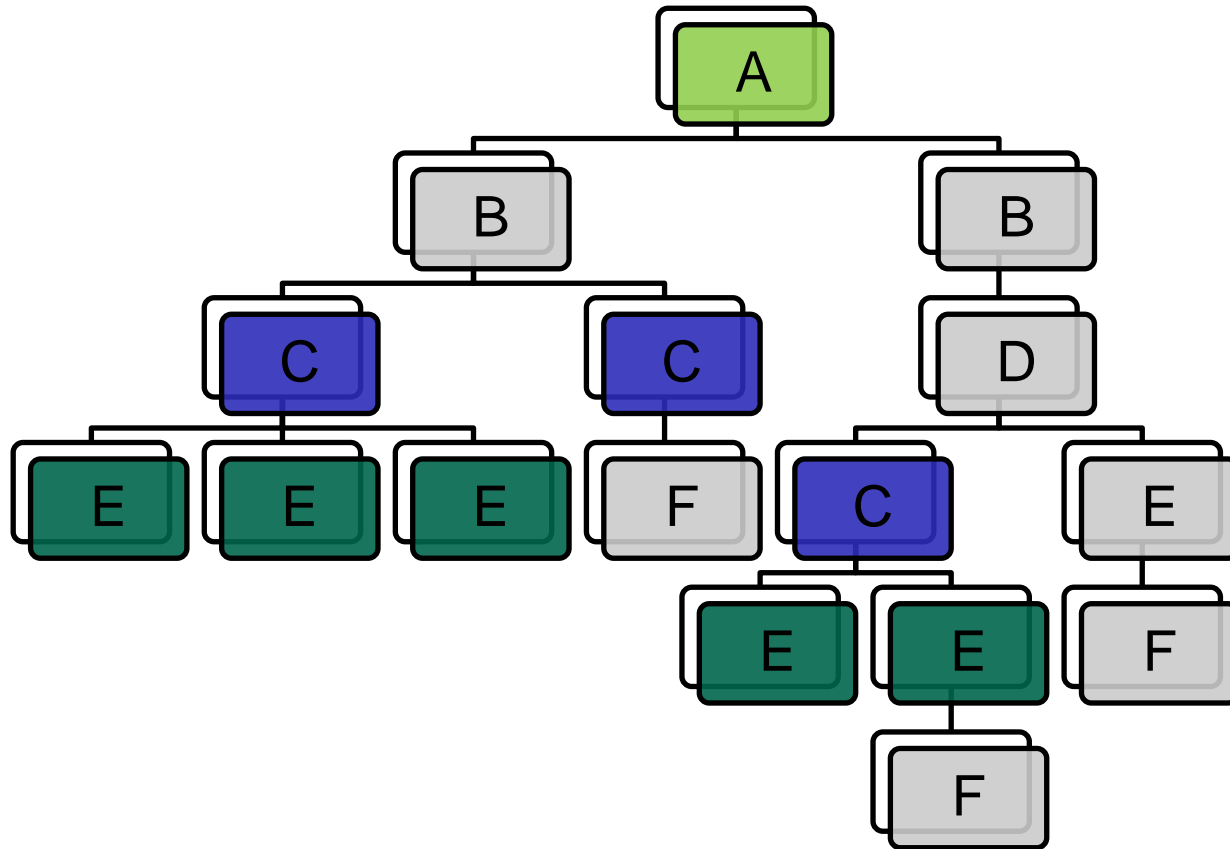
- A step maps a context into a set of nodes
- I.e. a step constructs a sequence (of nodes):
 - Each resulting node is used as the new context
 - and is replaced by the result of the mapping.
- The result of a step is used as the context of the next step.

Example



Example

descendant::C/child::E



Example

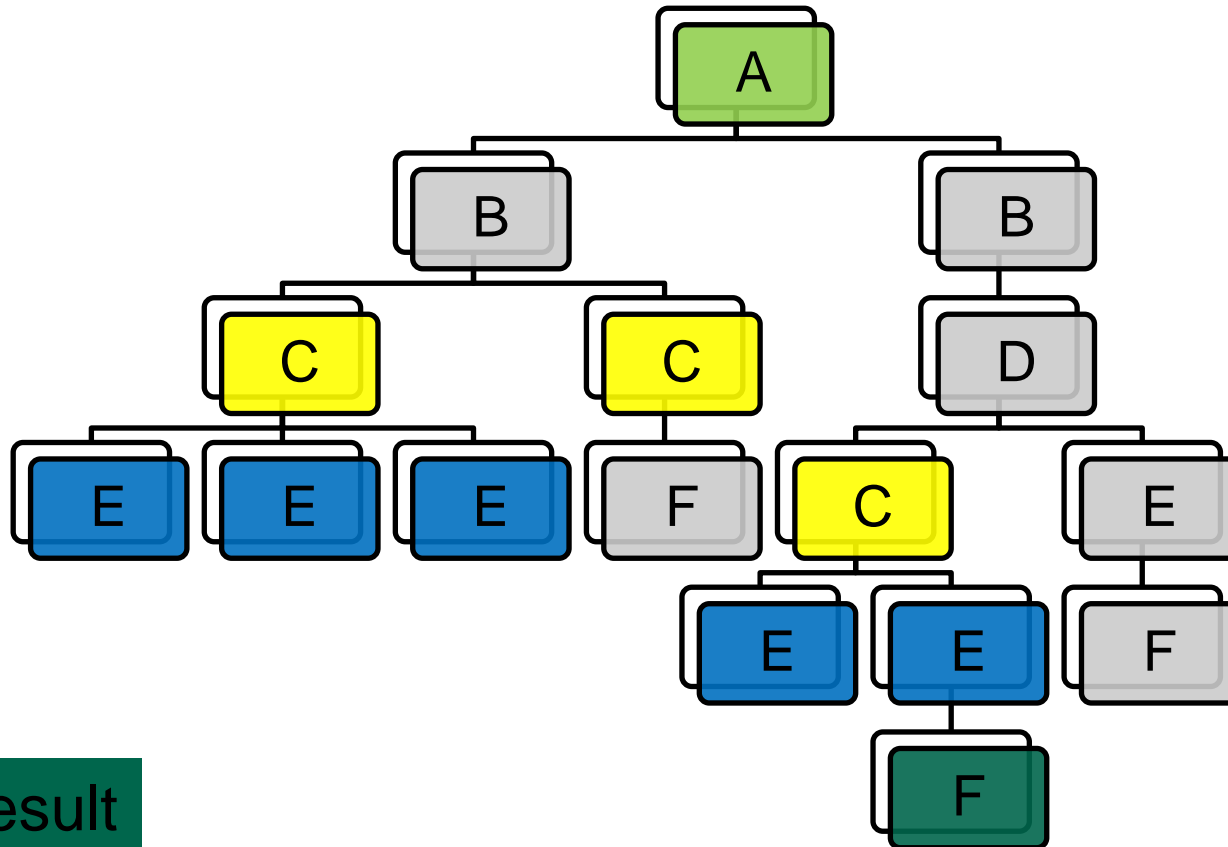
descendant::C/child::E/child::F

Kontext

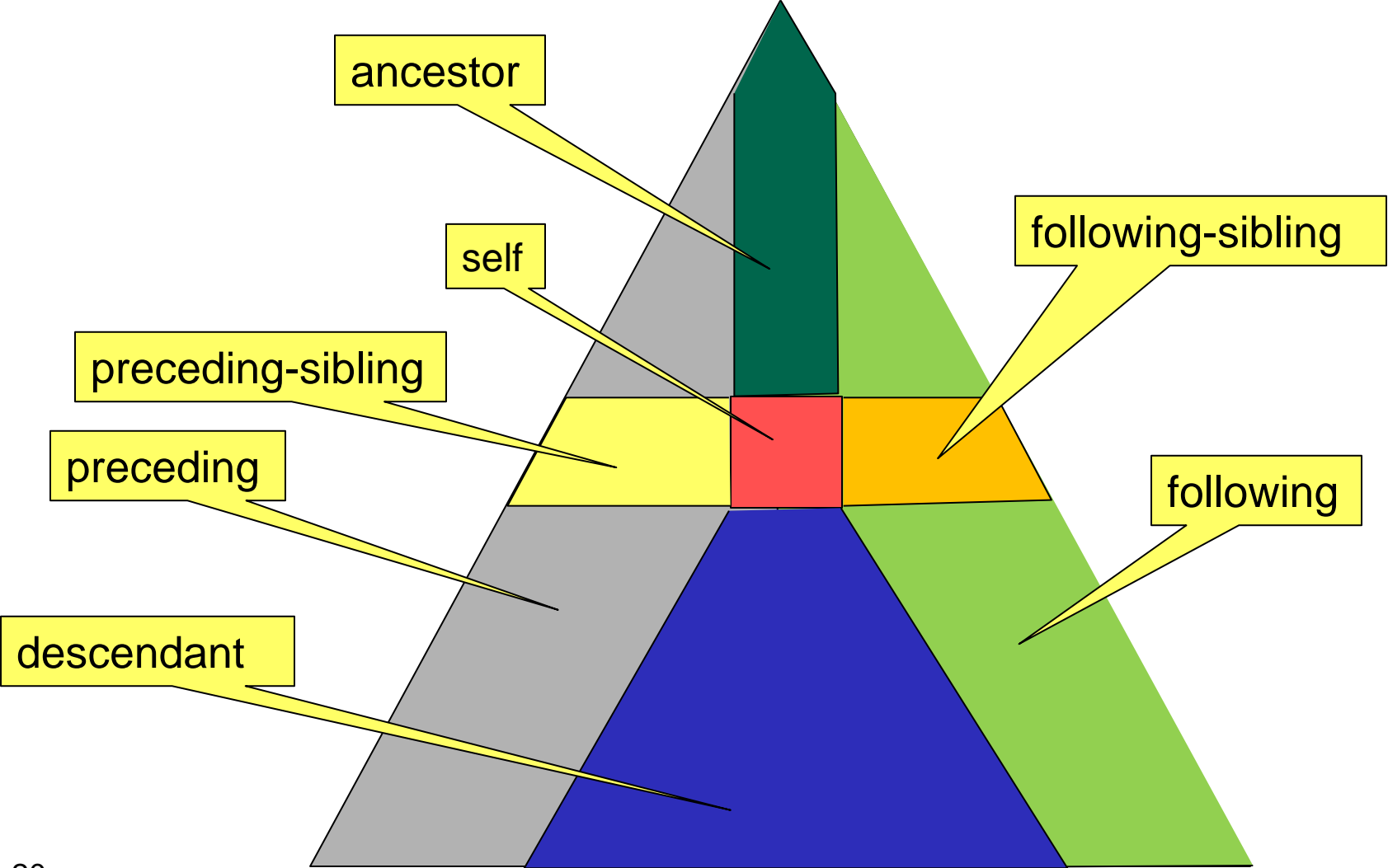
step1

step2

step3 / result



Axis



Node Test – Examples

- `text()`
- `comment()`
- `processing-instruction()`
- `node()`
- `*`
- `QName`
- `* : NCName`
- `NCName : *`

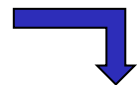
Atomization

- Atomization is applied to a value, when a specific data type is required.
- E.g. a sequence of nodes is transformed (atomized) into a sequence of strings.
- The default atomization:
 - Element nodes:
 - A concatenation of all “descendant“ nodes
 - For all other nodes:
 - The contents string

Expressions

- Path expressions
- Predicates
- Node tests
- Value comparison
- Sequence constructors
- Variable references
- Literal expressions
- and others ... e.g. ...

```
(10 to 40)[. mod 5 = 0 and position()>20]
```



```
30, 35, 40
```

XSLT

HTML and XML (1)

Example: Spaghetti Carbonara

Zutaten:

250 g [Spaghetti](#)

10 g [Butter](#)

100 g [Bacon](#)

3 [Eier](#)

30 g [Parmesan](#)

[Knoblauch](#)

[Pfeffer aus der Mühle](#)

[Salz](#)

3 Portionen - 625 Kcal pro Portion

*** - [20 min Zub.](#) - [20 min Ges.](#)*

Spaghetti Carbonara ist die klassische Zubereitung von Spaghetti mit frischen rohen Eiern und bedeutet soviel wie Spaghetti nach Köhler Art. Spaghetti Carbonara zählt zu den bekanntesten und beliebtesten Spaghettigerichten. Wichtig ist, dass man ganz frische Eier für Spaghetti Carbonara verwendet, da die Eier nur ganz leicht stocken dürfen - anders als Rührei - und fast roh gegessen werden. Spaghetti in reichlich Salzwasser bissfest kochen. Inzwischen Butter in einer [Pfanne](#) auslassen, aber nicht braun werden lassen. Bacon oder Bauchspeck in feine Streifen schneiden und in der heißen Butter knusprig ausbraten. Eine größere Zehe Knoblauch pellen und im Ganzen zum Bacon in die Pfanne geben. Wenn die Knoblauchzehe leicht zu bräunen



HTML and XML (2)

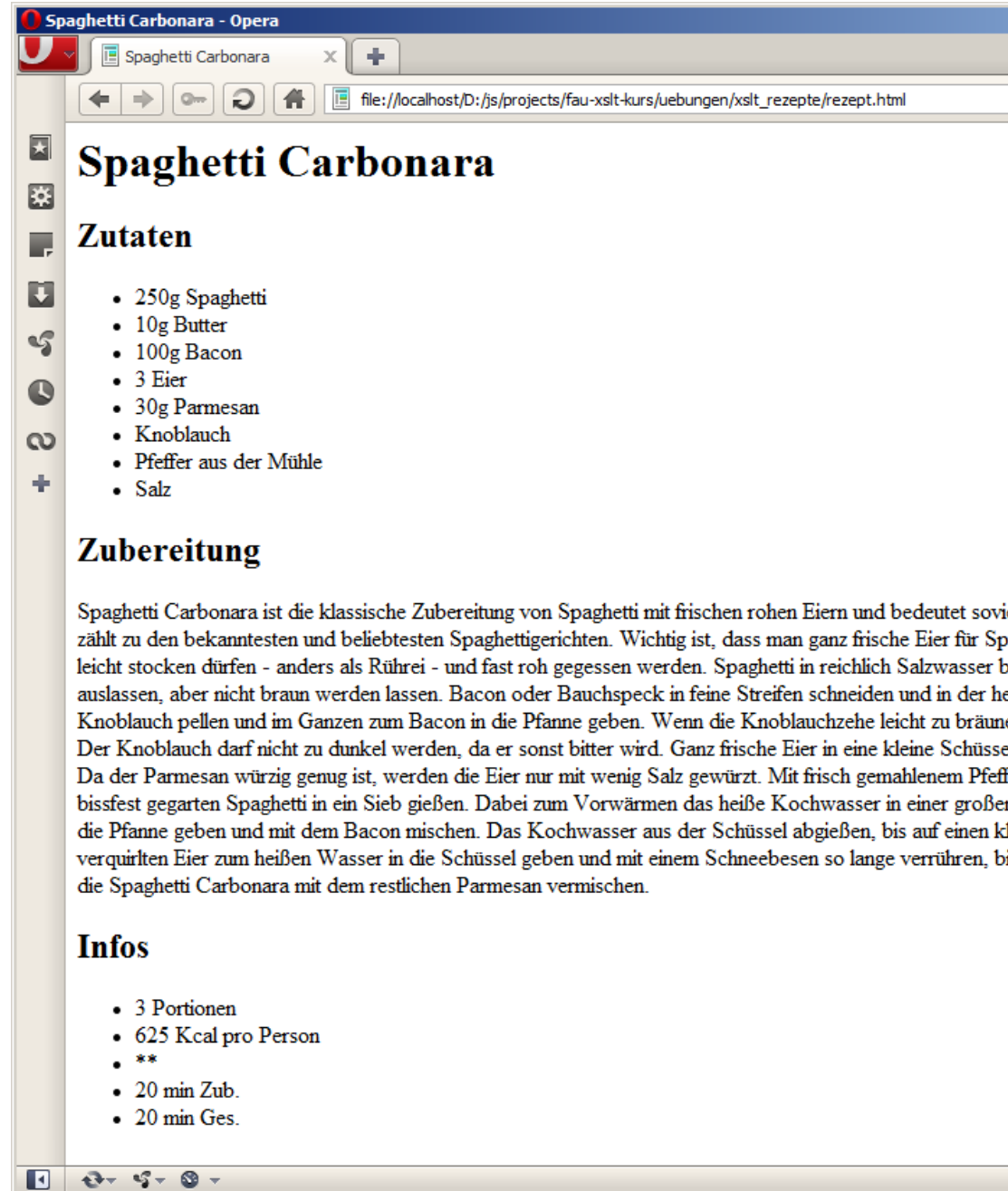
```
<?xml version="1.0" enco
<rezept>
<titel>Spaghetti Carbona
<zutaten>
  <item gewicht="250g">S
  <item gewicht ="10g">B
  <item menge="3">Eier</
  <item>Knoblauch</item>
</zutaten>
<zubereitung>
Spaghetti Carbonara ist
<zubereitung/>
<infos>
  <schwierigkeit>2</schw
  <dauer min="20" arbeit
  <dauer min="20" arbeit
</infos>
</rezept>
```



XSLT

```
<html>
<head><title>Spaghetti Carbona
<body>
<h1>Spaghetti Carbonara</h1>
<h2>Zutaten</h2>
<ul>
  <li>250g Spaghetti
  <li>10g Butter</ul>
<h2>Zubereitung</h2>
<p>Spaghetti Carbonara ist die
<h2>Infos</h2>
<ul>
  <li>**
  <li>20 min Zub.
  <li>20 min Ges.
</ul>
</body>
</html>
```

HTML and XML (3)



The screenshot shows a web browser window titled "Spaghetti Carbonara - Opera". The address bar displays the file path: `file:///localhost/D:/js/projects/fau-xslt-kurs/uebungen/xslt_rezepte/rezept.html`. The page content is as follows:

Spaghetti Carbonara

Zutaten

- 250g Spaghetti
- 10g Butter
- 100g Bacon
- 3 Eier
- 30g Parmesan
- Knoblauch
- Pfeffer aus der Mühle
- Salz

Zubereitung

Spaghetti Carbonara ist die klassische Zubereitung von Spaghetti mit frischen rohen Eiern und bedeutet somit zählt zu den bekanntesten und beliebtesten Spaghettigerichten. Wichtig ist, dass man ganz frische Eier für Sp leicht stocken dürfen - anders als Rührei - und fast roh gegessen werden. Spaghetti in reichlich Salzwasser b auslassen, aber nicht braun werden lassen. Bacon oder Bauchspeck in feine Streifen schneiden und in der he Knoblauch pellen und im Ganzen zum Bacon in die Pfanne geben. Wenn die Knoblauchzehe leicht zu bräun Der Knoblauch darf nicht zu dunkel werden, da er sonst bitter wird. Ganz frische Eier in eine kleine Schüssel Da der Parmesan würzig genug ist, werden die Eier nur mit wenig Salz gewürzt. Mit frisch gemahlenem Pfeff bissfest gegarten Spaghetti in ein Sieb gießen. Dabei zum Vorwärmen das heiße Kochwasser in einer großer die Pfanne geben und mit dem Bacon mischen. Das Kochwasser aus der Schüssel abgießen, bis auf einen kl verquirlten Eier zum heißen Wasser in die Schüssel geben und mit einem Schneebesen so lange verrühren, bi die Spaghetti Carbonara mit dem restlichen Parmesan vermischen.

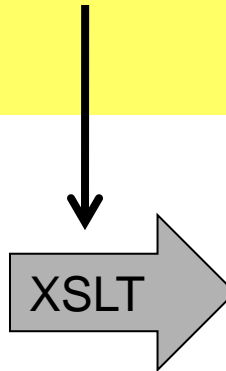
Infos

- 3 Portionen
- 625 Kcal pro Person
- **
- 20 min Zub.
- 20 min Ges.

An XSL Stylesheet

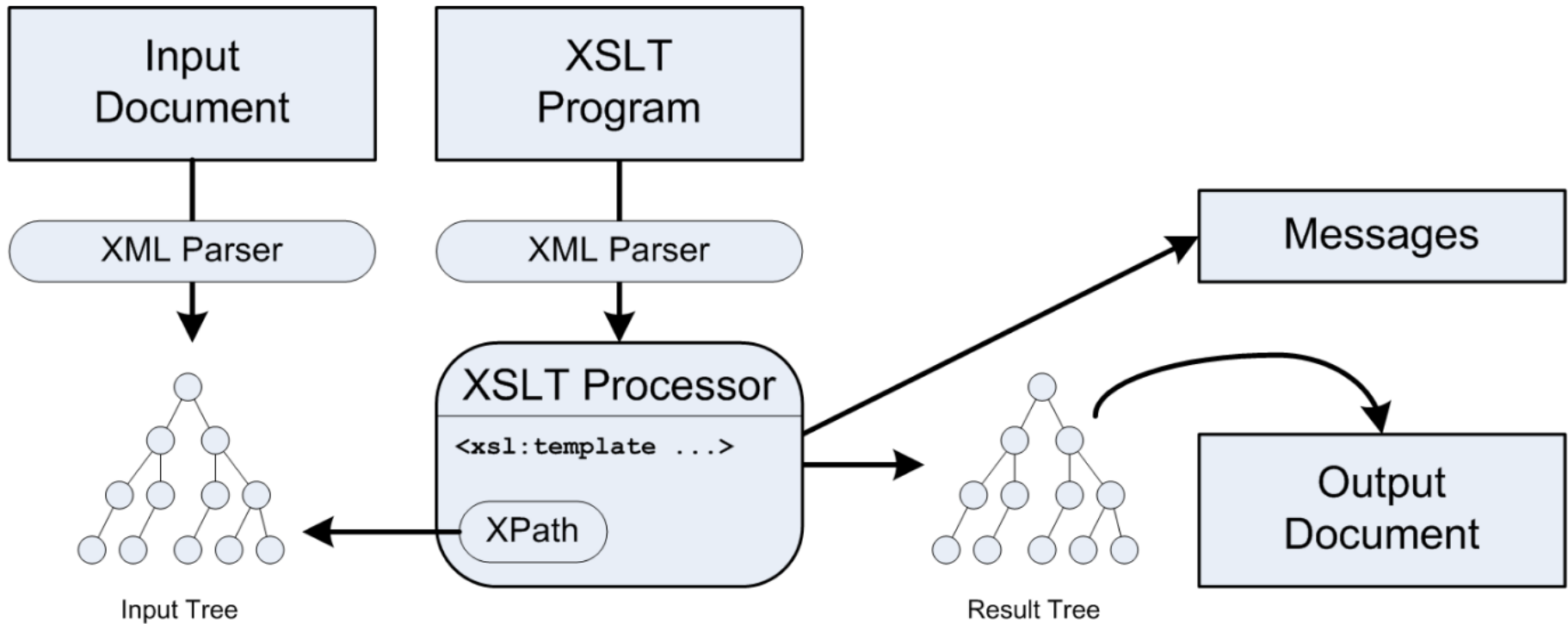
```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:template match="/">
    <html>
      <xsl:value-of select="rezept"/>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

```
<?xml version="1.0" encodi
<rezept>
<titel>Spaghetti Carbonara
<zutaten>
  <item gewicht="250g">Spa
  <item gewicht ="10g">But
  <item menge="3">Eier</it
  <item>Knoblauch</item>
</zutaten>
<zubereitung>
Spaghetti Carbonara ist di
```



```
<html>
  Spaghetti Carbonara
  Spaghetti
  Butter
  Eier
  Knoblauch
  Spaghetti Carbonara ist die
  klassische Zubereitung von
  Spaghetti mit
```

XSLT-Transformation



XSLT

- Eine Programmiersprache zur Verarbeitung und Erzeugung von Bäumen.
- Wichtigste Operationen:
 - Traversierung von Bäumen
 - Rekursiver Aufruf
 - Regeln für spezielle Teilbäume

Generating Output

- Generating Text
- Constructors for elements and attributes
- recursion
- modes
- conditionals
- repetition
- copying and sorting nodes
- named templates
- XSLT functions
- grouping

Tools

Eclipse

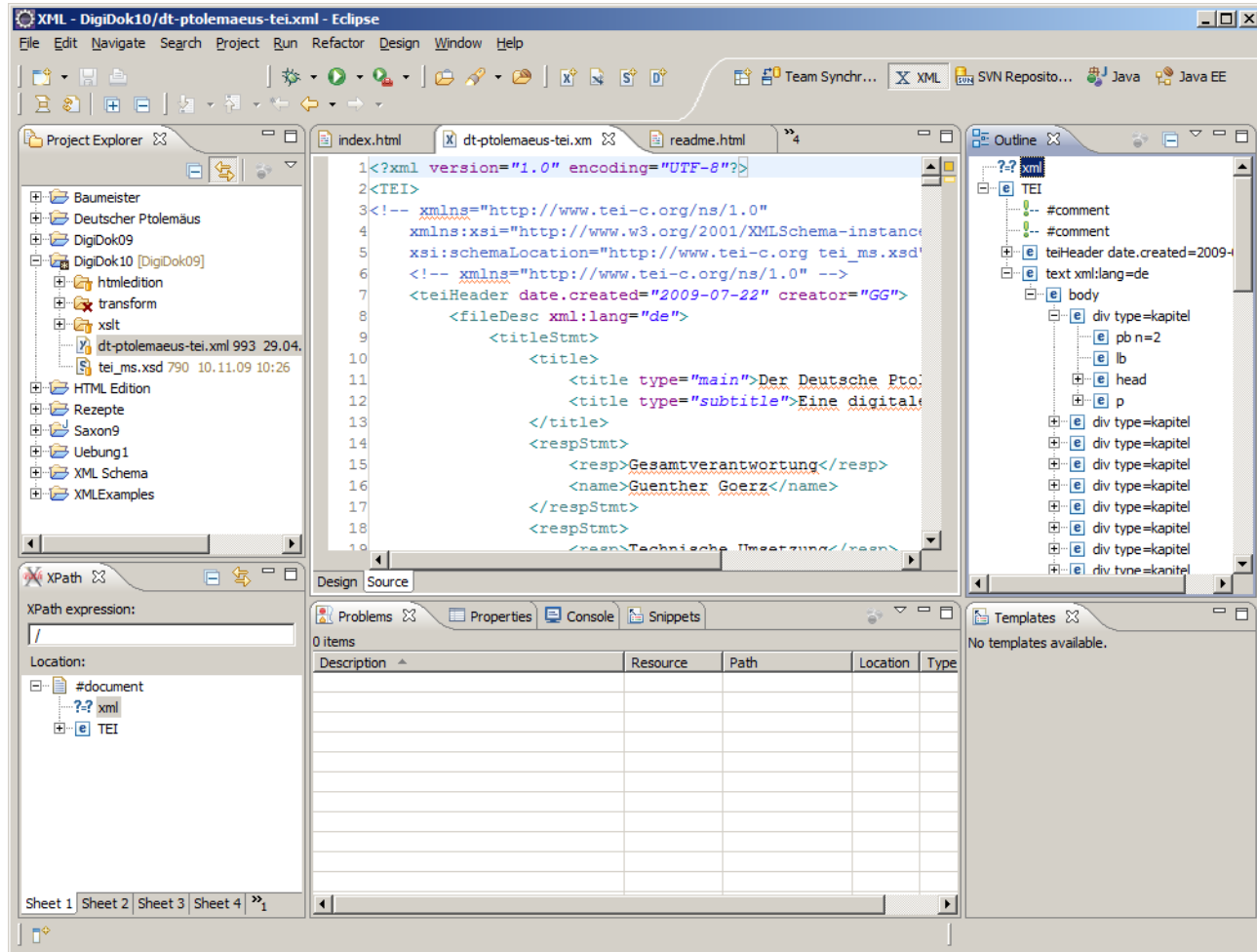
Cocoon

eXist

... and others

Eclipse

- Eclipse
- Syntax support for XML, CSS, JavaScript, and XHTML
- Input completion
- XSLT transformation
- XSLT debugging
- No royalties

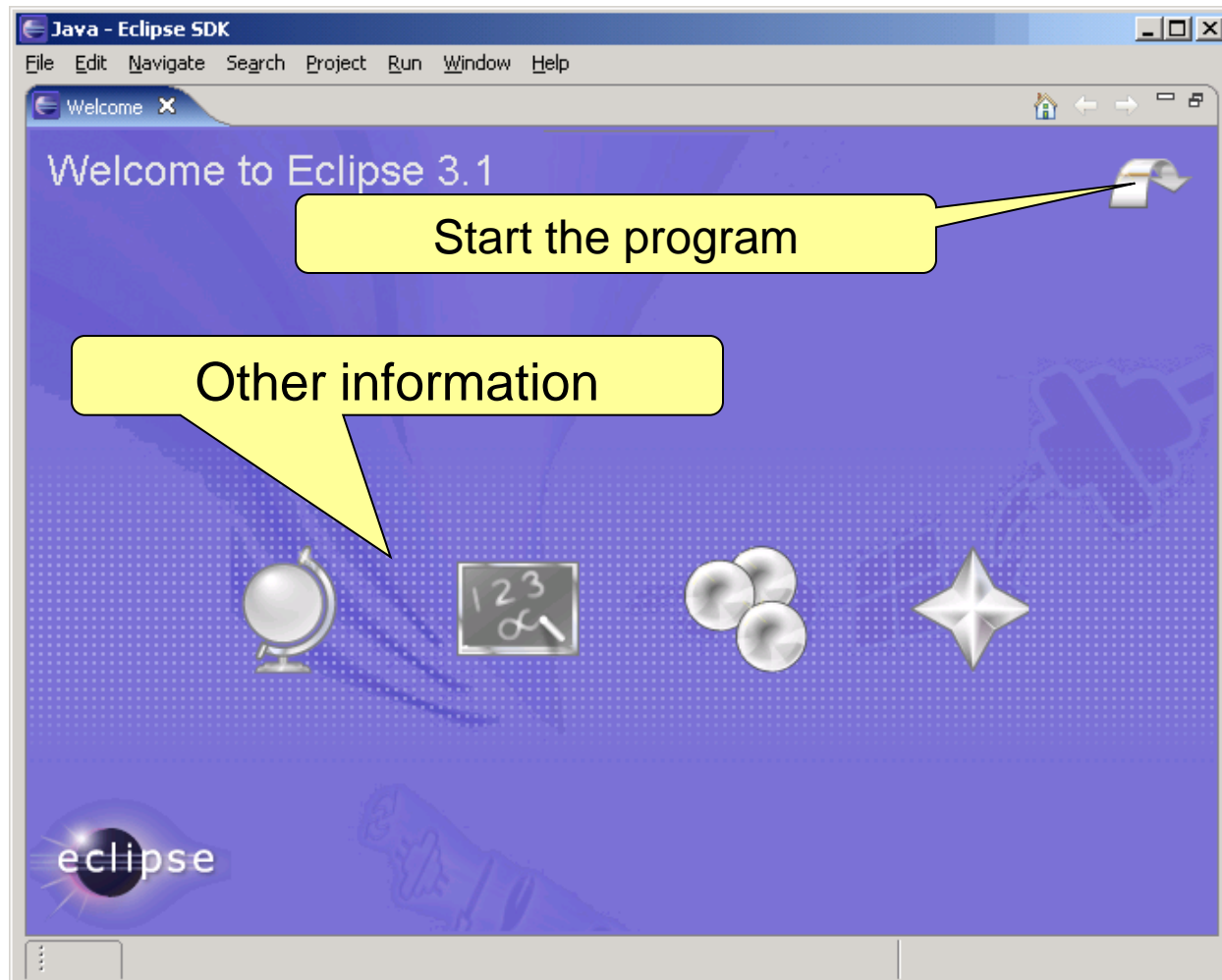


<http://www.eclipse.org/downloads/> - Java EE Developers

Jump Start with Eclipse

- Google: Eclipse download
- Get the J2EE developer packages (includes XML processing)
- Choose the right platform (Mac / 32 / 64bit / ...)
- Unzip the download (no other installation required – besides Java)
- Start Eclipse

Eclipse Startup



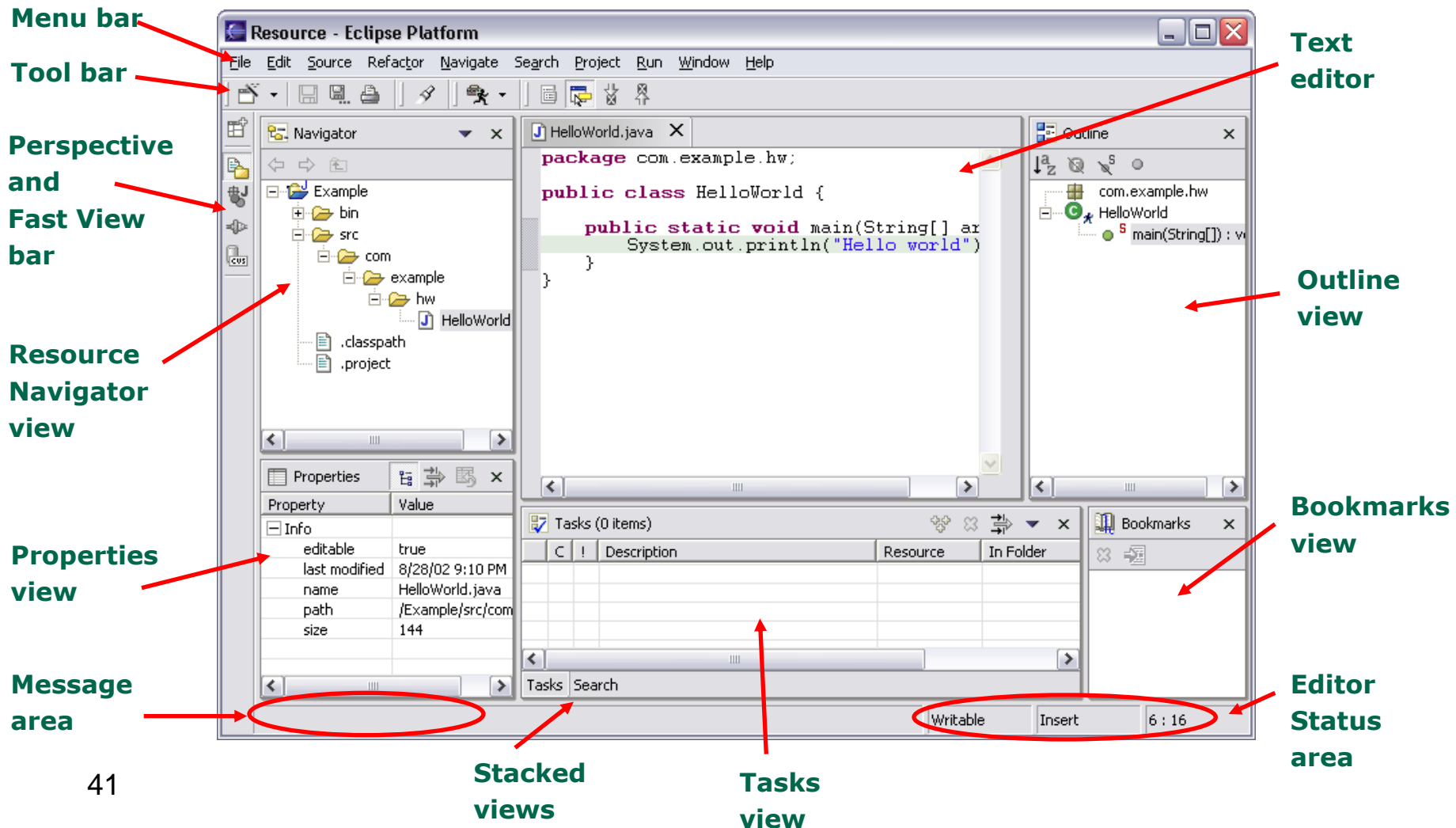
Workspace

- When starting Eclipse it occupies a “workspace”
- A “workspace” is a folder for multiple projects.
- Workspaces may be changed.
- Typically, a workspace is used for similar projects.
- Workspace configurations
 - Go to preferences
 - Find a page supplying a search string
 - Change the workspace character encoding to UTF-8
 - Display line numbers in the editor

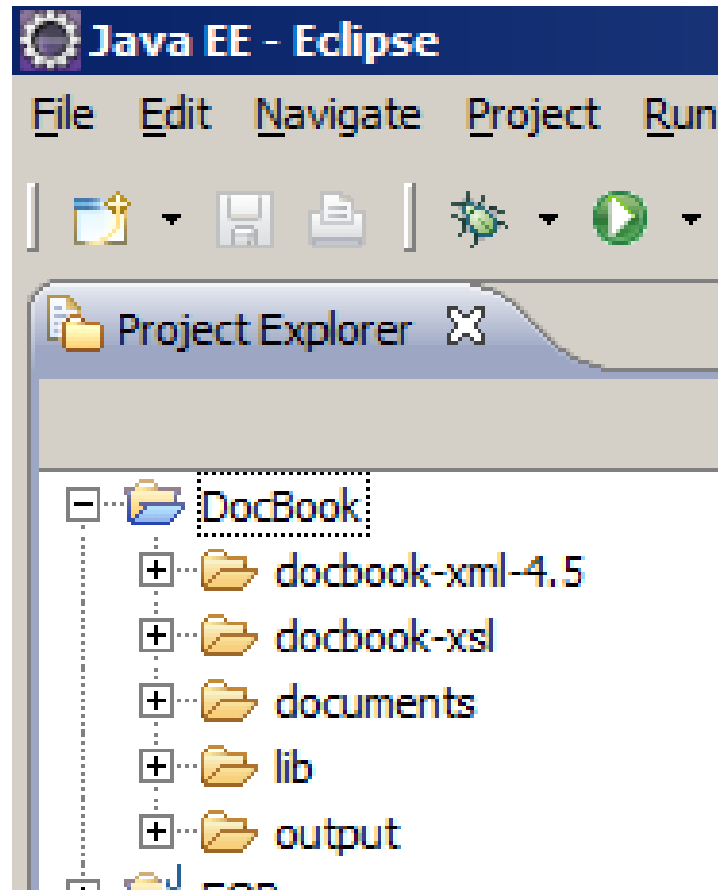
Projects

- A project is used for a particular context or task
- A project is a folder, that may be moved later.
- There are different project types
 - Java projects
 - Web projects
 - etc.
- Projects may be opened or closed.
- Projects are part of a workspace.

Workbench Terminology



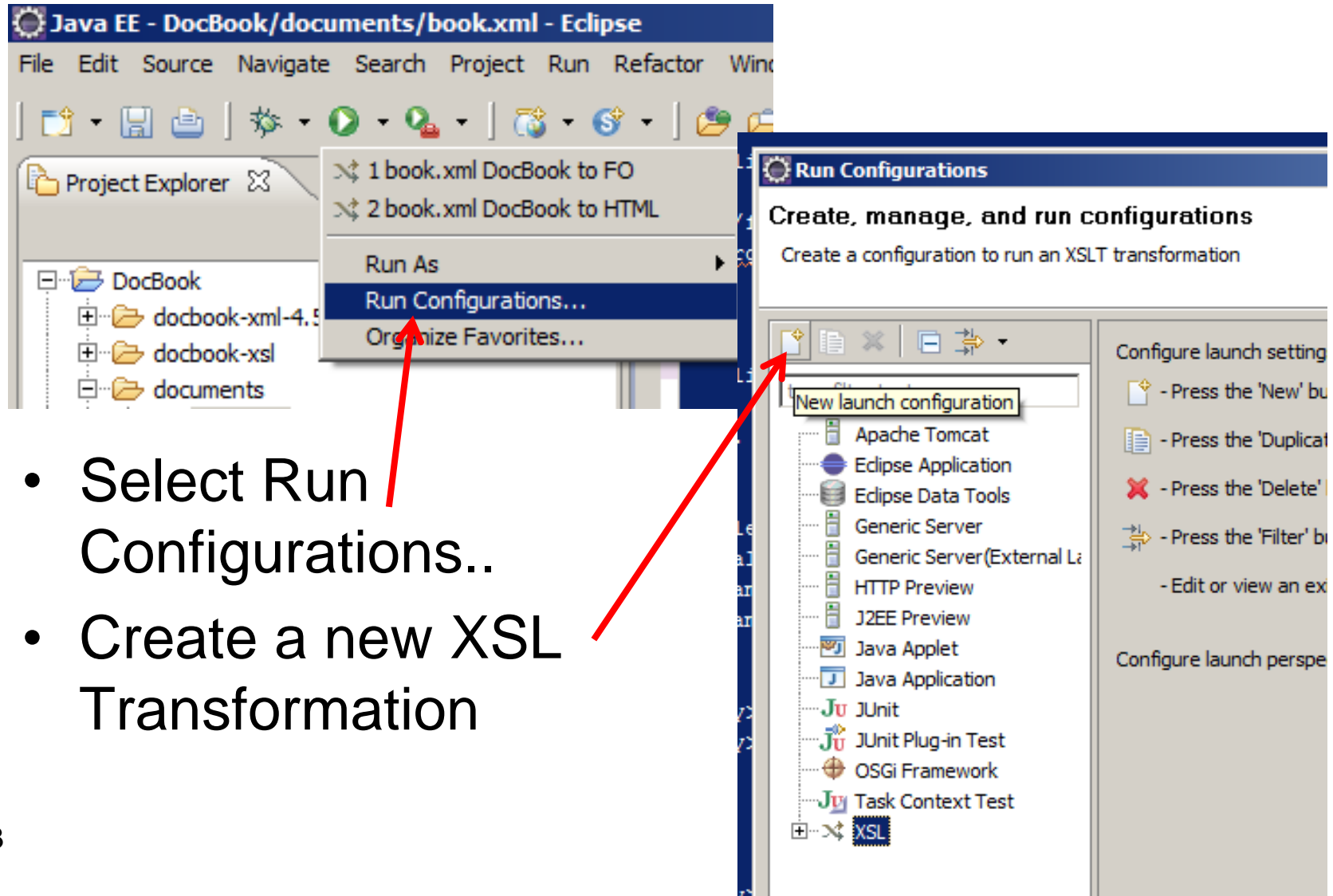
Organize your Project



E.g.:

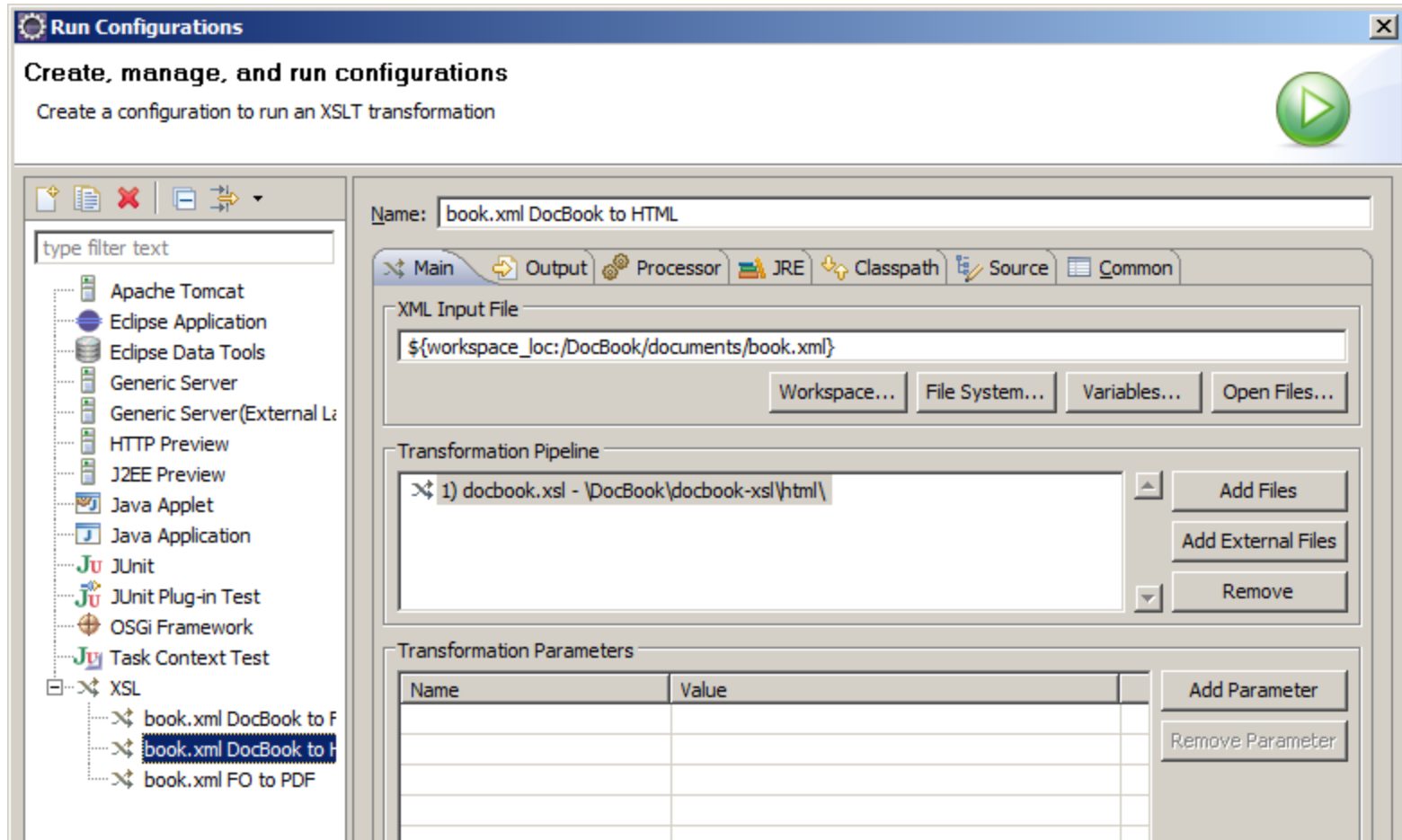
- DocBook DTD
- DocBook Stylesheets
- Own documents
- Libraries (e.g. Xalan and Saxon)
- Output documents

Run an XSL Transformation

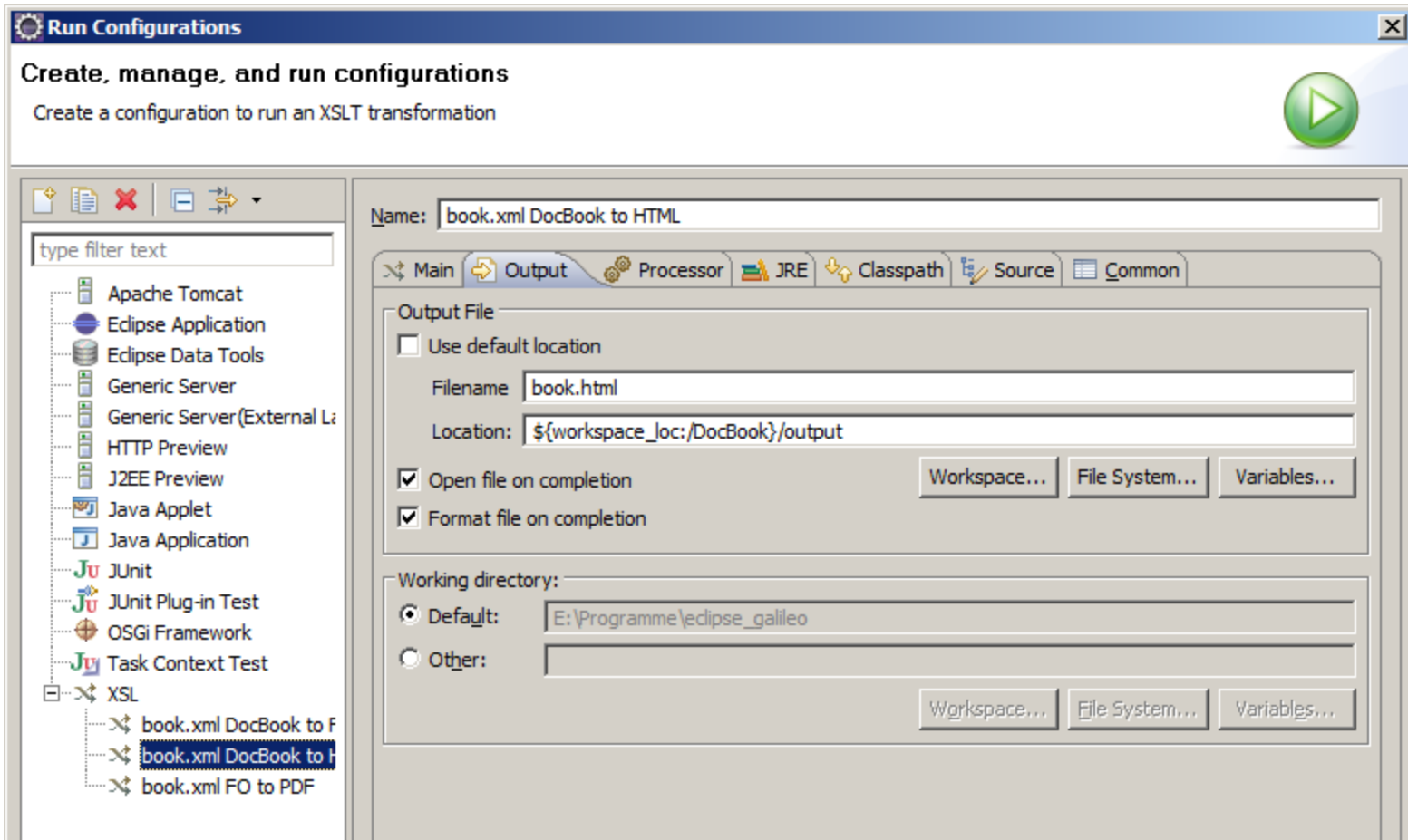


- Select Run Configurations..
- Create a new XSL Transformation

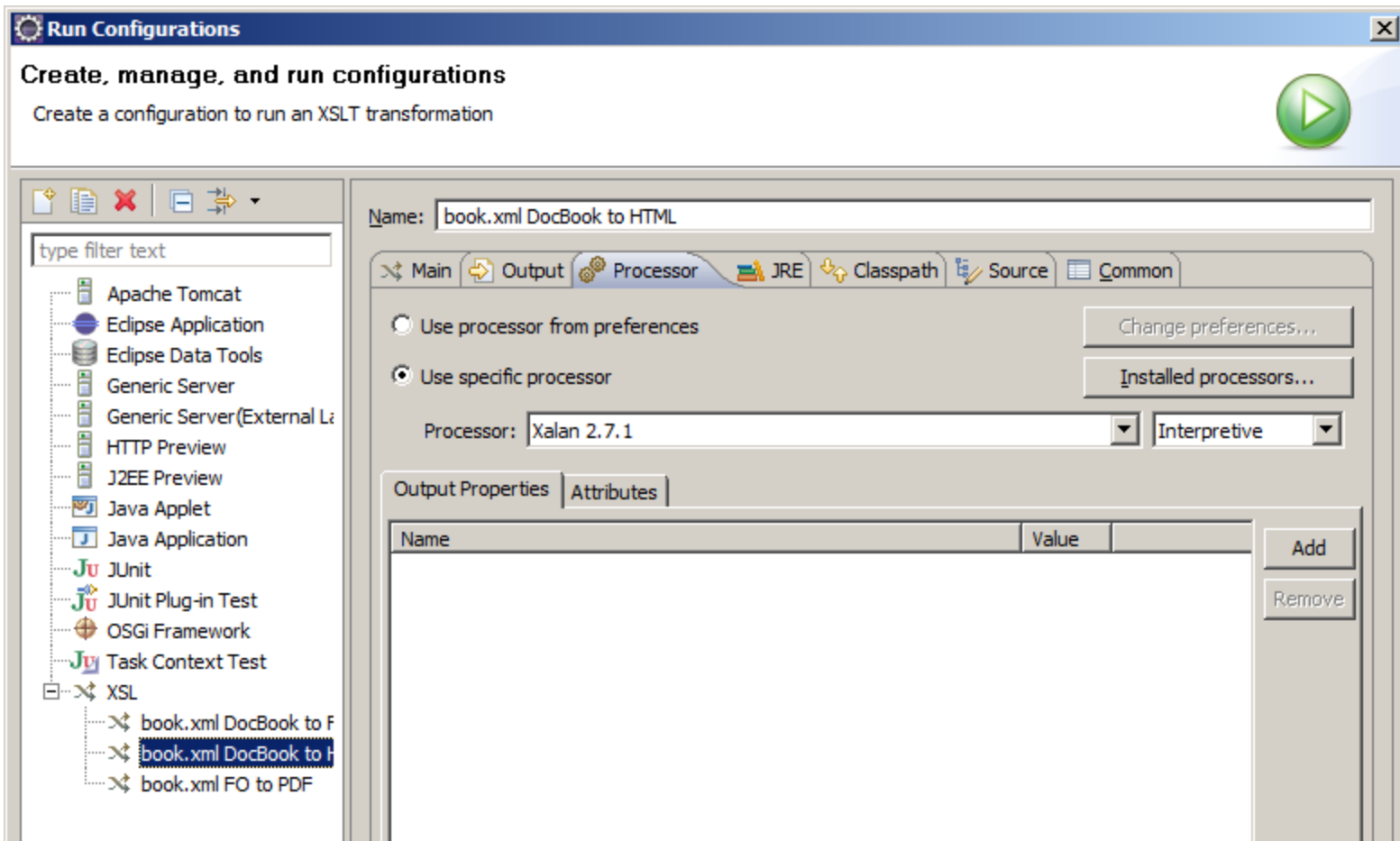
XSLT Run Configuration (1)



XSLT Run Configuration (2)



XSLT Run Configuration (3)



Code Completion

- Based on XML schema or DTD

The screenshot shows the Eclipse IDE interface with the 'Java EE - DocBook/documents/book.xml' file open. The Project Explorer on the left shows the project structure, including folders like 'docbook-xml-4.5', 'docbook-xsl', 'documents', 'lib', 'output', and files like 'book.xml'. The main editor displays XML code with the following structure:

```
<surname>Schneeberger</surname>
</author>
</articleinfo>
<sect1 label="1.0">
  <title>DocBook und Eclipse</title>
  <para>
    Ein erster Absatz.
  </para>
  <
  <
```

A code completion popup is visible, listing various XML elements. The 'blockquote' element is highlighted. To the right of the popup, a tooltip provides details for the selected element:

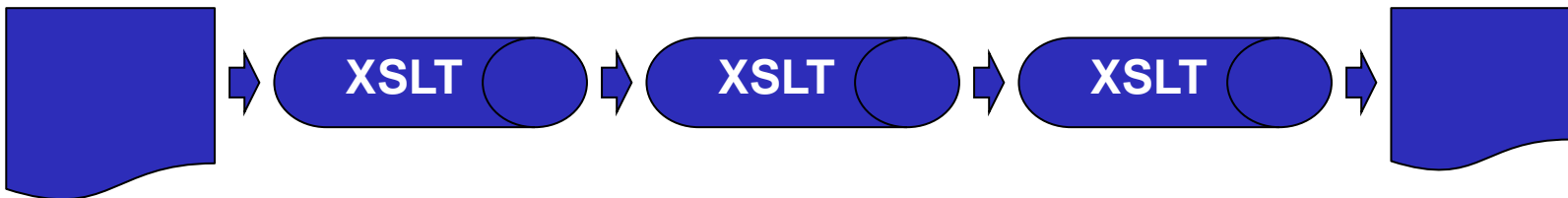
Element: blockquote
doc:A quotation set off from the main text.

Content Model: (blockinfo?, title?, attribution?, (calloutlist | glosslist | bibliolist | itemizedlist | orderedlist | segmentedlist | simplelist | variablelist | caution | important | note | tip | warning | literallayout | programlisting | programlistingco | screen | screenshot | screenshot | synopsis | cmdsynopsis | funcsynopsis | classsynopsis | fieldsynopsis | constructorsynopsis | destructorsynopsis | methodsynopsis | formalpara | para | simpara | address | blockquote | graphic | graphicco | mediaobject | mediaobjectco | informalequation | informalexample | informalfigure | informaltable | equation | example | figure | table | msgset | procedure | sidebar | ...

Cocoon and eXist

Apache Cocoon

- A web application framework
- Central processing method: XSLT
- Pipelining and “Separation of concerns”



Getting started with Cocoon/eXist

- I used eXist-1.2.6-rev9165.war on top of tomcat 6
- Include eXist-1.2.6-rev9165.war in the ...\\webapps\\ folder of tomcat
- Fire up tomcat with ...\\bin\\tomcat6.exe
- run /Applications/eXist/bin/startup.sh (or something similar)
- Direct the browser to: <http://localhost:8080/exist/> or <http://localhost:8080/exist-1.2.6-rev9165/>
- eXist:
 - Go to Administration > Admin
 - login admin, pw: (empty)
 - Browse Collections > Create Collection (new: dtpt)
 - Select dtpt, Choose File : ../dt-ptolemaeus-tei.xml

Import XML into eXist

- I had a problem with the local schema file
- Import was not possible
 - See log files:
...\\Tomcat 6.0\\webapps\\eXist\\WEB-INF\\logs\\
- Remove
`xsi:schemaLocation="http://www.tei-c.org/ns/1.0 dt-ptolemaeus-tei.xsd"`
from the root element.

Local Folders (JS)

- Eclipse Workspace:
d:\js\projects\digidok\eclipse_ws\
d:\js\projects\digidok\workspace
- Project Files:
e:\Programme\Apache Software
Foundation\Tomcat 6.0\webapps\eXist\dtpt1\
e:\Programme\Apache Software
Foundation\Tomcat 6.0\webapps\dtpt1\
- Safety-Copy:
d:\js\projects\digidok\eXistCocoon\
d:\js\projects\digidok\workspace
- eXist configuration:
d:\js\projects\digidok\eXistCocoon\sitemap.xmap
d:\js\projects\digidok\workspace

Hello Ptolemaeus (1)

- Create a folder
...\\webapps\\eXist\\dtpt
 - Application spaces/folders may be created just inside of exist
 - These folders may contain XML files but also XQuery, XSLT etc.
- Add a pipeline to cocoon
 - Edit file eXist\\sitemap.xmap
 - Add

```
<map:pipeline>  
  <map:match pattern="dp0">  
    <map:generate src="xmldb:exist:///db/dtpt/dt-ptolemaeus-tei.xml"/>  
    <map:serialize type="xml"/>  
  </map:match>  
</map:pipeline>
```
- Take care: Do not name an eXist folder the same as a pipeline!

Cocoon Pipeline

- Pattern: The name (pattern) of the pipeline to call it from the browser. Eg.:
`http://localhost:8080/eXist/dtpt`
- Generate: Source of data
- Transform: used for XQuery or XSLT transformation steps
- Serialize: Always the final step ... defaults to HTML. Add `type="xml"` for XML

Add Saxon9b for XSLT2 (1)

- See: <http://wiki.apache.org/cocoon/Saxon>
- Get Saxon9b from the internet
- add the jar file to ...\\eXist\\WEB-INF\\lib\\
- Comment in Saxon support in ...\\eXist\\WEB-INF\\cocoon.xconf

```
<component logger="core.xslt-processor"  
  role="org.apache.excalibur.xml.xslt.XSLTProcessor/saxon"  
  class="org.apache.excalibur.xml.xslt.XSLTProcessorImpl">  
  <parameter name="use-store" value="true"/>  
  <parameter name="incremental-processing" value="false"/>  
  <parameter name="transformer-factory" value="net.sf.saxon.TransformerFactoryImpl"/>  
</component>
```

Add Saxon9b for XSLT2 (2)

- Add a new transformer for saxon to ...\\eXist\\sitemap.xmap

```
<map:transformer name="xslt-saxon" pool-grow="2" pool-max="32" pool-min="8",  
  src="org.apache.cocoon.transformation.TraxTransformer">  
  <use-request-parameters>false</use-request-parameters>  
  <use-browser-capabilities-db>false</use-browser-capabilities-db>  
  <xslt-processor-role>saxon</xslt-processor-role>  
</map:transformer>
```

Hello Ptolemaeus (2)

- Update cocoon pipeline
 - Edit file eXist\sitemap.xmap
 - Update

```
<map:pipeline>
  <map:match pattern="dp1">
    <map:generate src="dtpt1/dt-ptolemaeus-tei.xml"/>
    <map:transform type="xslt-saxon-wparam" src="dtpt1/select1.xsl">
  <map:parameter name="use-request-parameters" value="true"/>
</map:transform>
  <map:transform type="xslt-saxon" src="dtpt1/div2flat.xsl"/>
  <map:transform type="xslt-saxon" src="dtpt1/flat2page.xsl"/>
  <map:transform type="xslt-saxon" src="dtpt1/select2.xsl"/>
  <map:transform type="xslt-saxon" src="dtpt1/pages2html.xsl"/>
  <!-- map:serialize type="xml"/ -->
  <map:serialize/>
</map:match>
</map:pipeline>
```

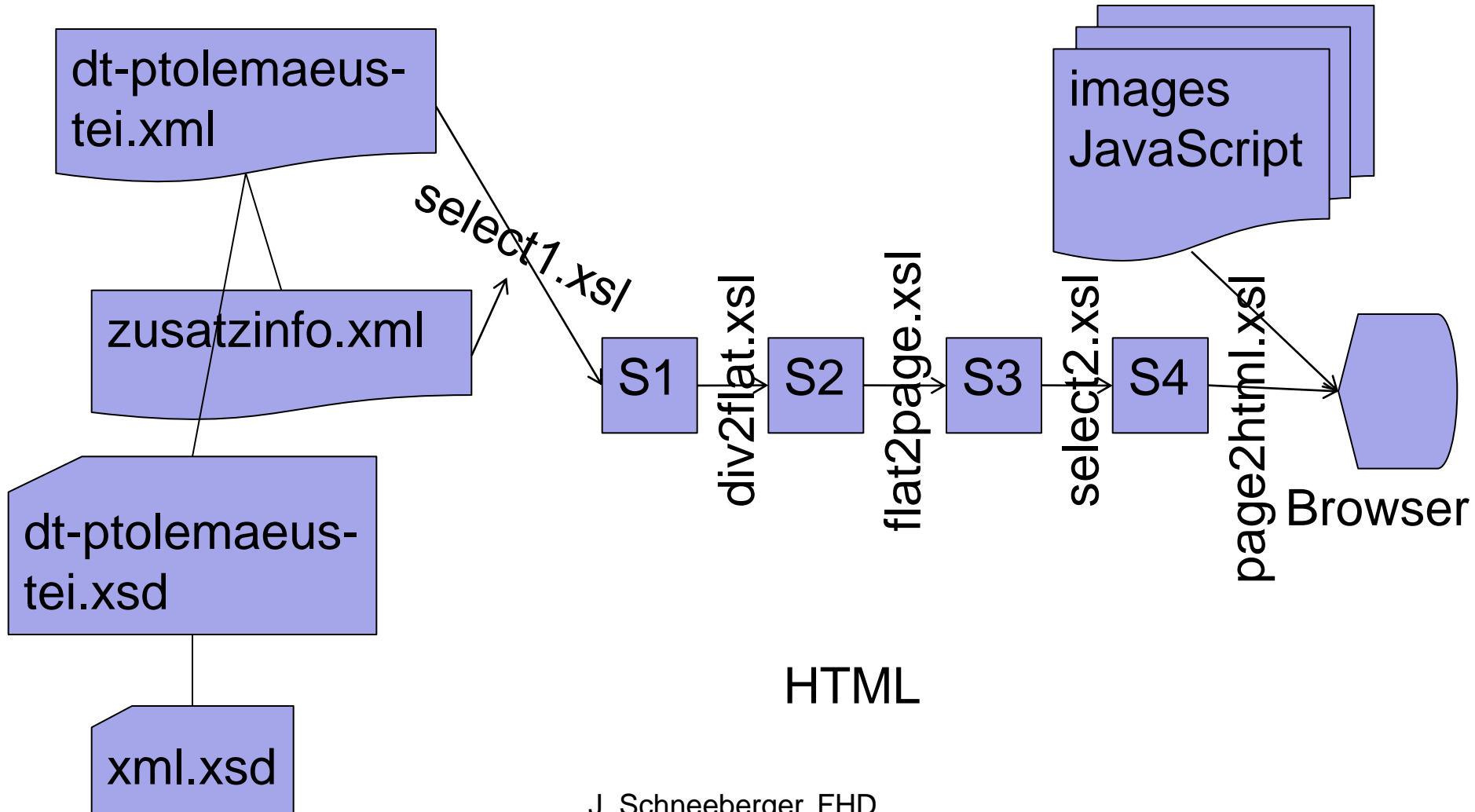
Testing / intermediate results

```
<map:pipeline>  
  <map:match pattern="dp2">  
    <map:generate src="dtpt1/dt-ptolemaeus-tei.xml"/>  
    <map:transform type="xslt-saxon-wparam" src="dtpt1/select1.xsl">  
      <map:parameter name="use-request-parameters" value="true"/>  
    </map:transform>  
    <map:transform type="xslt-saxon" src="dtpt1/div2flat.xsl"/>  
    <map:transform type="xslt-saxon" src="dtpt1/flat2page.xsl"/>  
    <map:transform type="xslt-saxon" src="dtpt1/select2.xsl"/>  
    <map:serialize type="xml"/>  
  </map:match>  
</map:pipeline>
```



Comment out if
desired

Ptolemaeus neu



select1.xsl

- Erzeugt ein Zwischendokument s1 mit dem Wurzelelement `<root>` mit dem Inhalt:
 - Infos aus *tei:teiHeader*
 - Die Navigations-Elemente `<navi>`
 - Einträge zum Inhaltsverzeichnis `<toc>`
 - Der Inhalt *tei:text*
 - Die Zusatzinfos in `<info>` aus *zusatzinfo.xml*

Passing Parameters

```
<!-- NOTE: Added by JS -->
<map:transformer logger="sitemap.transformer.xslt" name="xslt-saxon"
  pool-grow="2" pool-max="32" pool-min="8"
  src="org.apache.cocoon.transformation.TraxTransformer">
  <use-request-parameters>false</use-request-parameters>
  <use-browser-capabilities-db>false</use-browser-capabilities-db>
  <xslt-processor-role>saxon</xslt-processor-role>
</map:transformer>
<map:transformer logger="sitemap.transformer.xslt" name="xslt-saxon-wparam"
  pool-grow="2" pool-max="32" pool-min="8"
  src="org.apache.cocoon.transformation.TraxTransformer">
  <use-request-parameters>true</use-request-parameters>
  <use-browser-capabilities-db>false</use-browser-capabilities-db>
  <xslt-processor-role>saxon</xslt-processor-role>
</map:transformer>
```

Further Considerations

Further Tasks

- Milestone transformations
- Printed output
- Display of terminology (places, persons, ...)
- Integrate commentary
- Add revision control