TEI Processing Model

TEI Annual Conference and Members' Meeting 2025

Workshop outline

- Rationale behind TEI Processing Model
  - XSLT: duplicate code for different output formats, e.g. TEI Stylesheets
  - Case study: FRUS (3.5k lines of code per output format vs 350 lines of ODD)
- Data vs Presentation or Encoding vs visualization
- Case studies
  - dateline, choice with abbr/expan, persName & person note
- Very quick introduction to XPath
  - XML as a tree; element and attribute nodes; parent/child, ancestor/descendant, root
  - filters
  - Sequences
- Main concepts of the TEI Processing Model
  - Typical behaviours and content parameter adjustments
  - Styling via cssClass or tei-* classes
- Hands-on session: the ODD editor

https://github.com/eeditiones/workshop

Setting-up eXist-DB and package installation

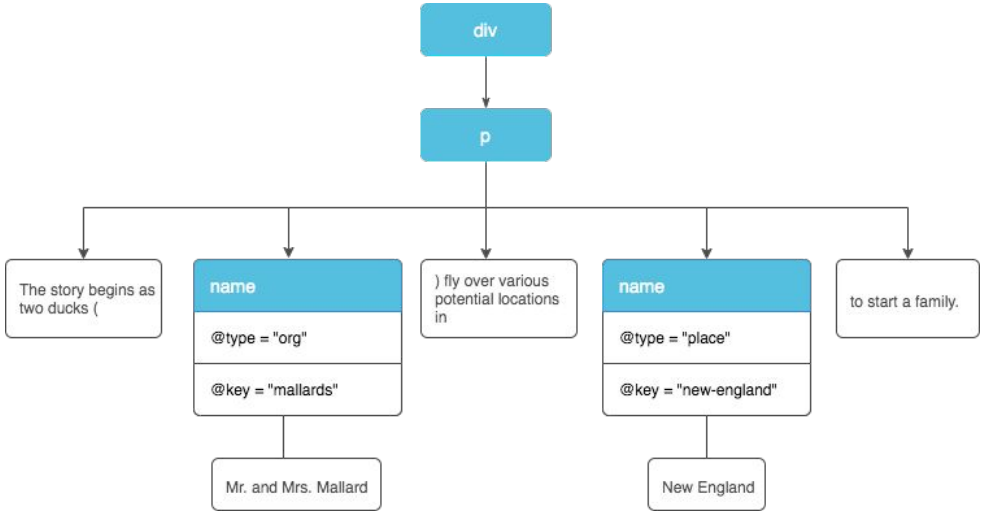
Package installation

- Run eXist-DB
- Open http://localhost:8080/exist/apps/dashboard/index.html
- Log in as **`admin`** and click on package manager.
- Click on the add package symbol in the upper left corner and select the **`.xar`** file contained in **`workshop/data/xars`**

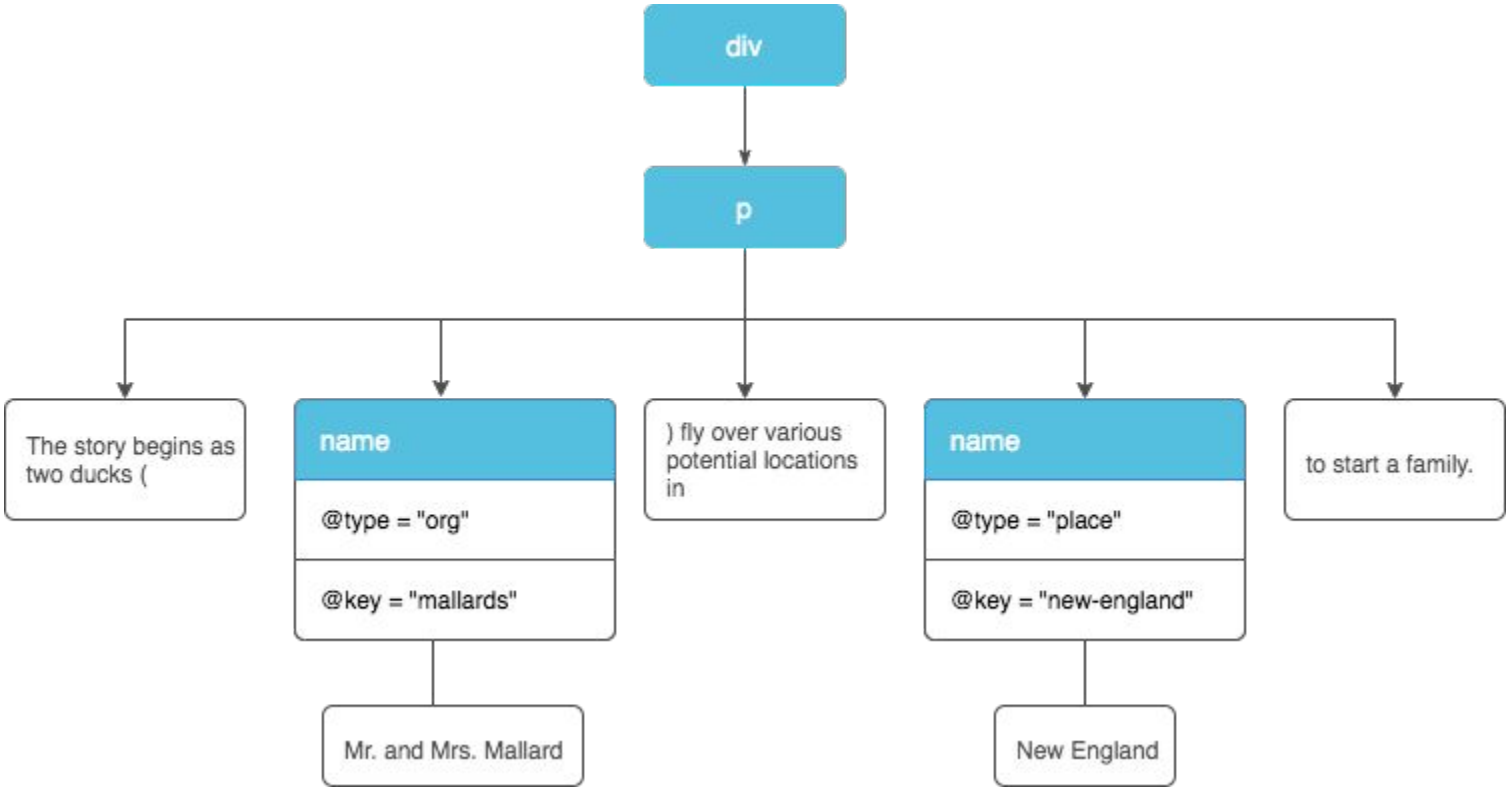
Introduction to XPath

XML as a tree

```
<?xml version="1.0" encoding="UTF-8"?>
 <div xmlns="http://my.fantasy.namespace">
 <!-- This whole document is in a made-up
namespace -->
      <p>The story begins as two ducks
     (<name type="org" key="mallards">Mr.
     and Mrs. Mallard</name>)fly over
     various potential locations in <name
     type="place" key="new-england">New
     England</name> to start a family.</p>
 </div>
```



XML as a tree

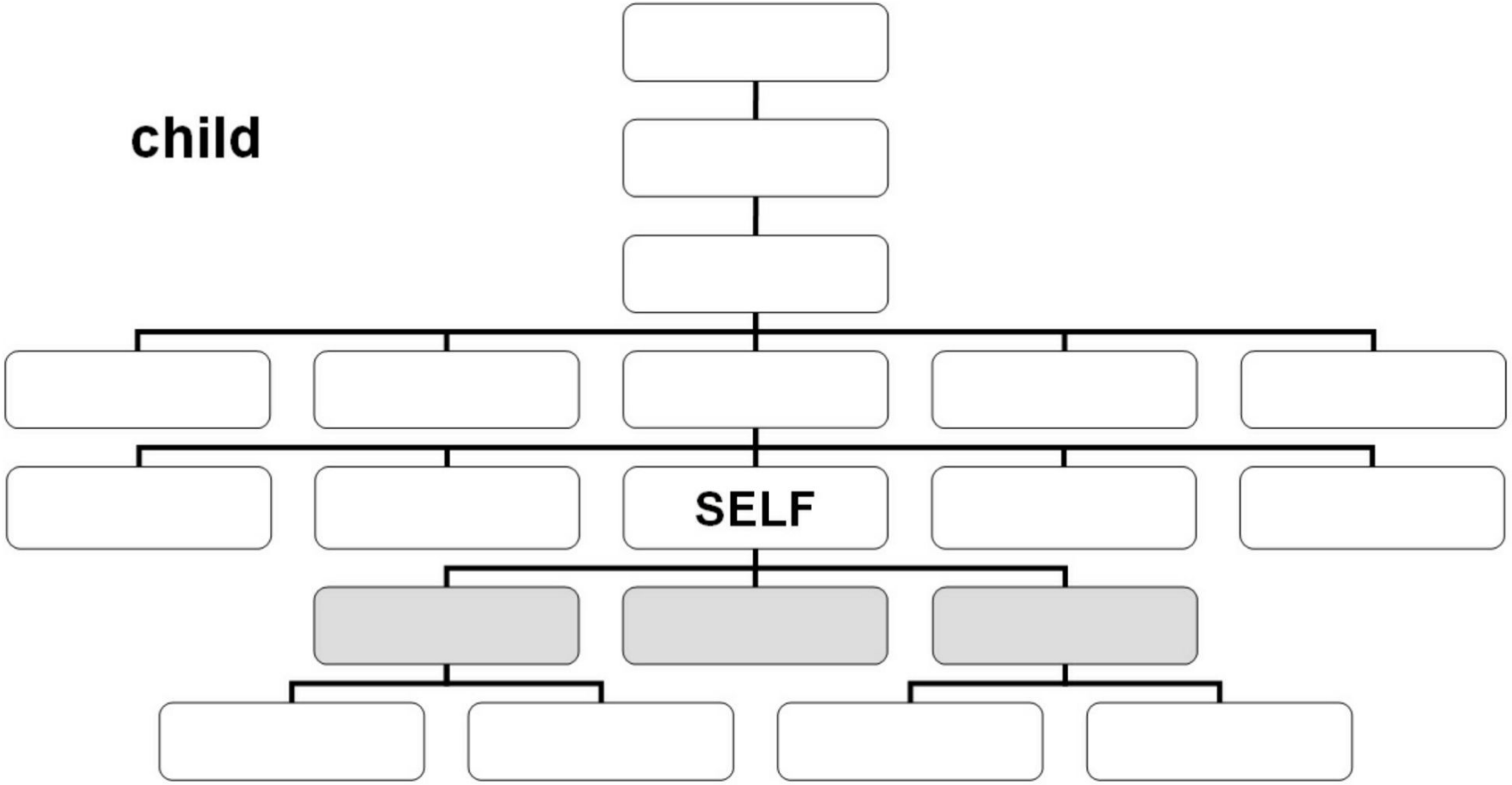

div
p
The story begins as two ducks (
name
@type = "org"
@key = "mallards"
Mr. and Mrs. Mallard
) fly over various potential locations in
name
@type = "place"
@key = "new-england"
New England
to start a family.

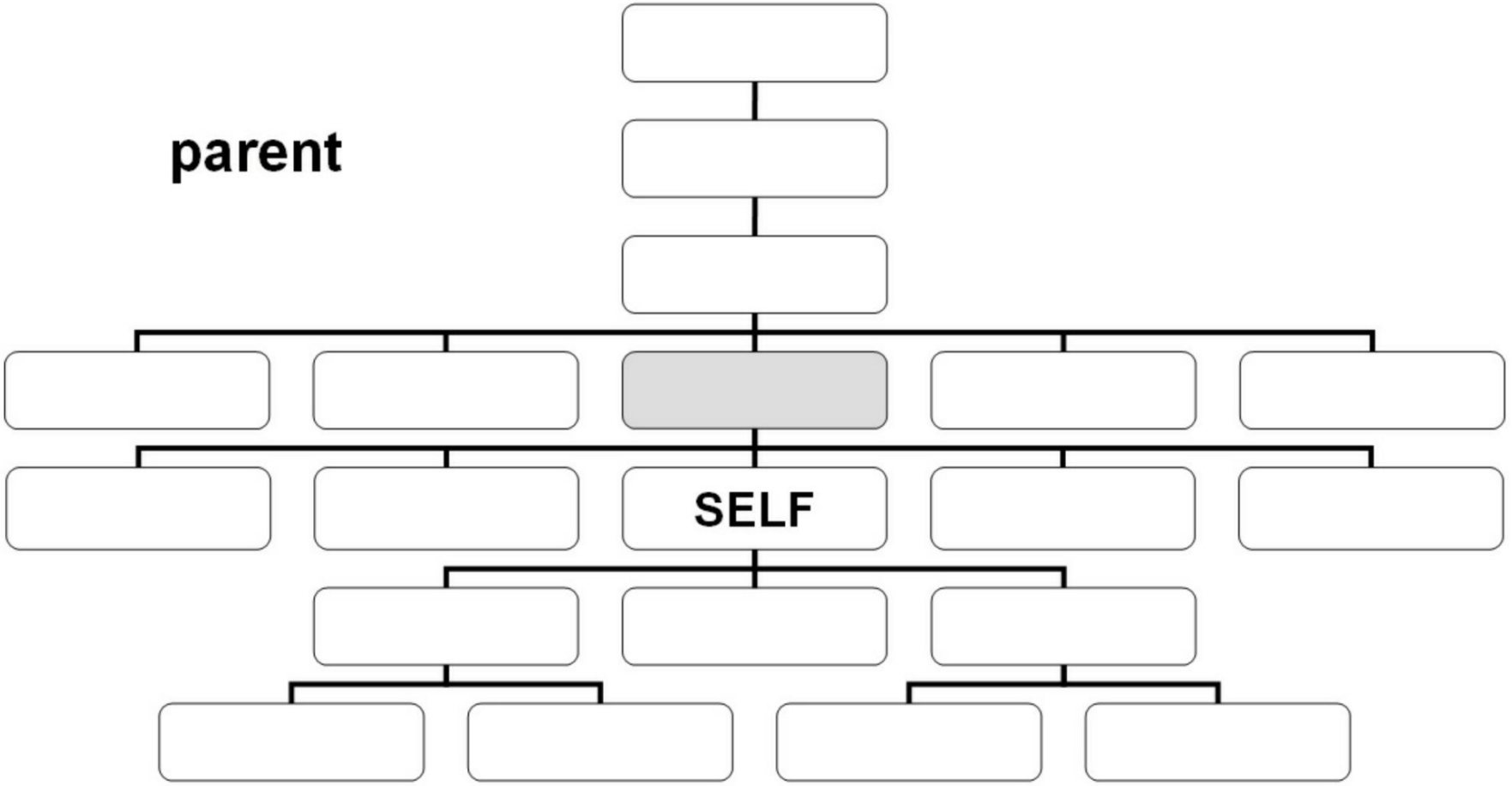
Node types in XML document

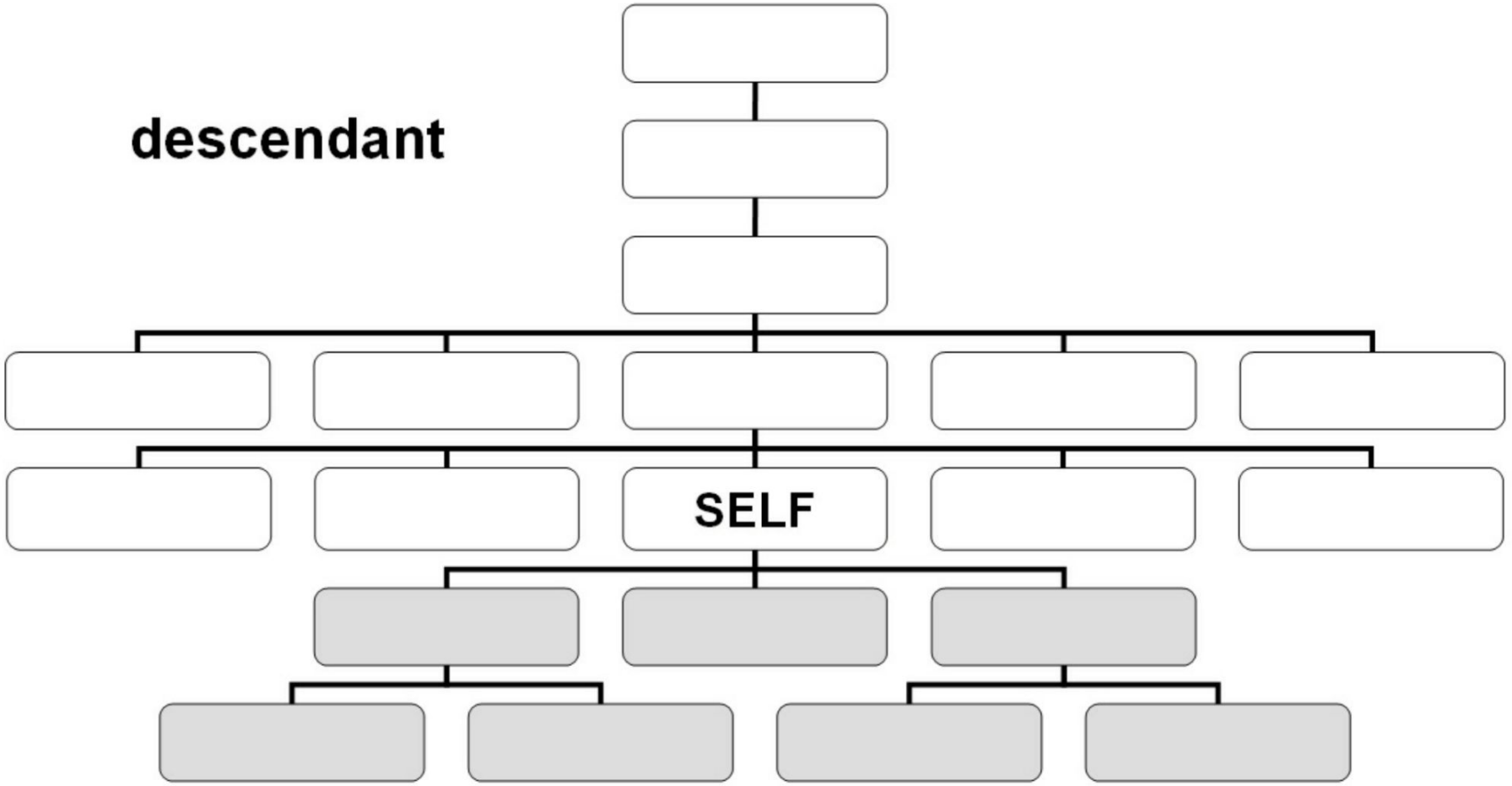
- element
- attribute
- text
- namespace
- processing-instruction
- comment
- document node

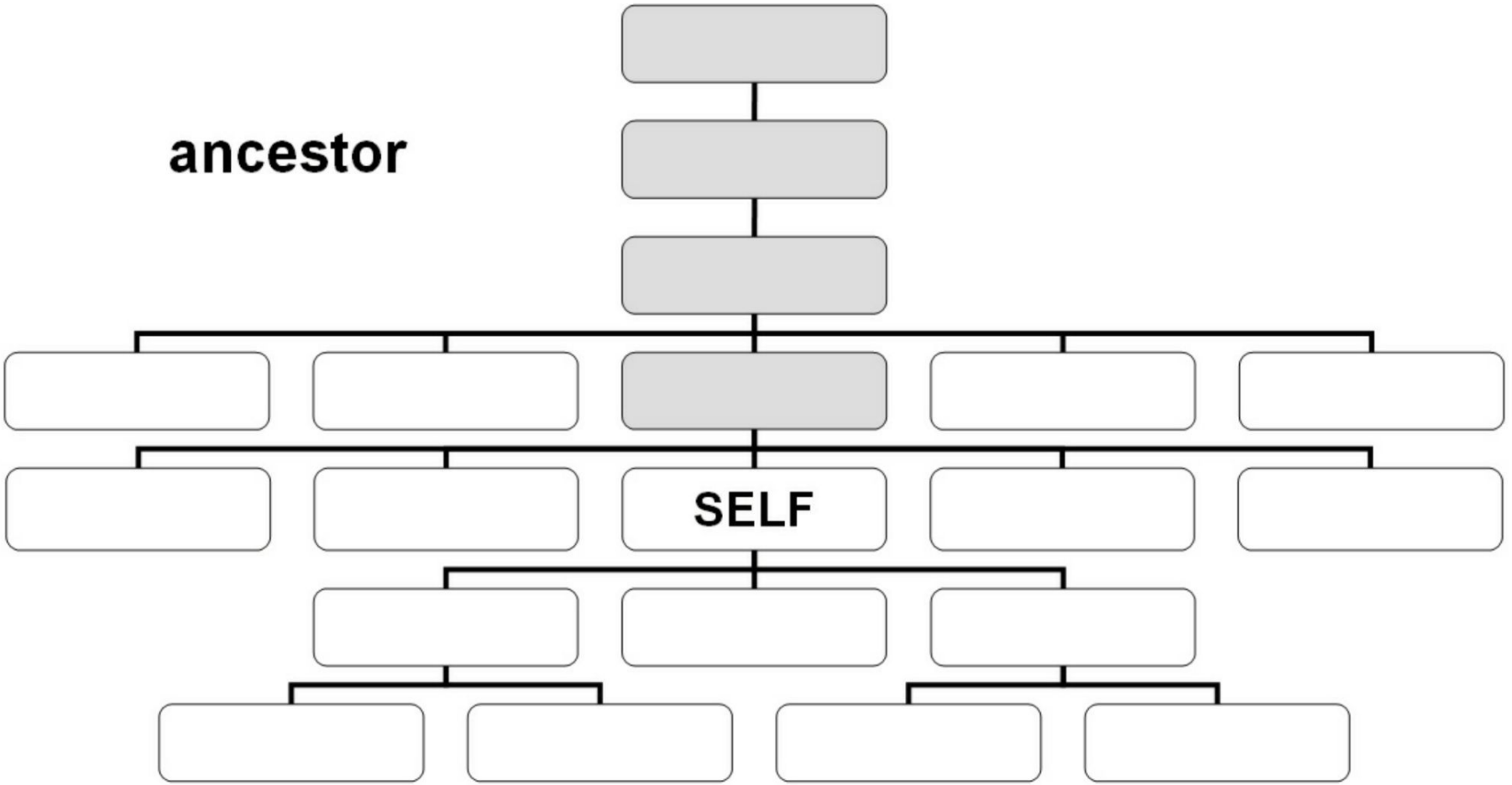
Family relationships between nodes

- parent
- child
- ancestor
- descendant
- sibling
- self


child
SELF


parent
SELF


descendant
SELF

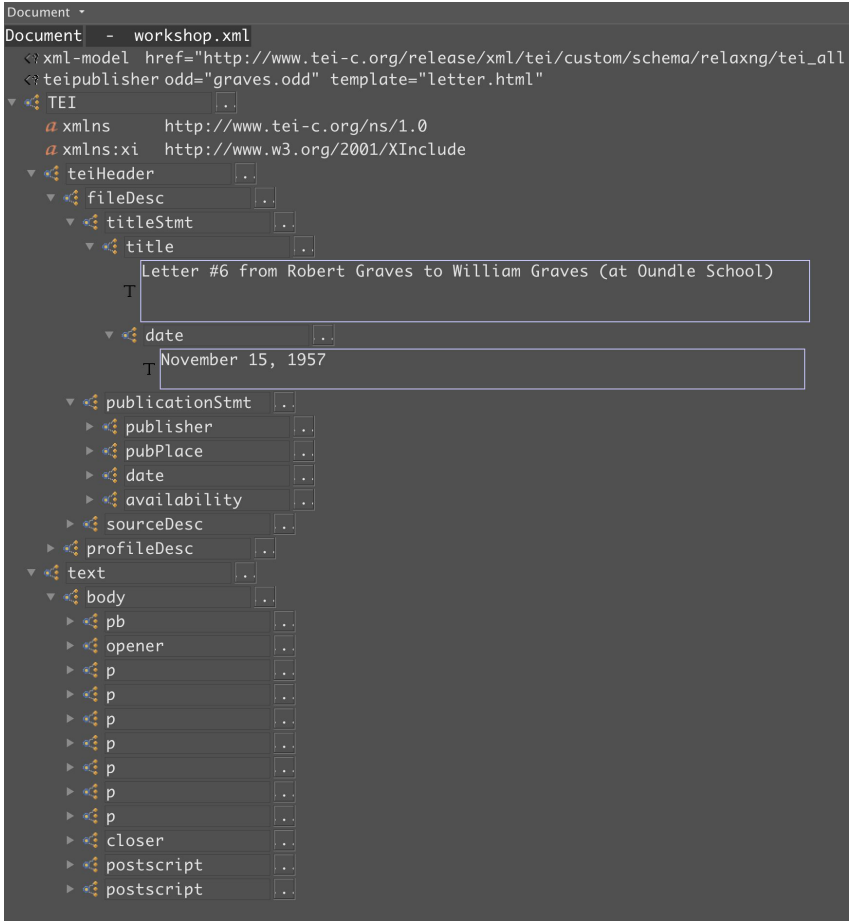

ancestor
SELF

XPath

- XPath is a language for selecting sets of nodes
- Main thing an XPath expression does is to describe, in a formal way that a computer can process easily, certain sets of nodes, e.g.:
  - title of the document
  - all first paragraphs in a section
  - all references to a name of a person
  - all sections that are not of type *appendix*

```
<TEI xmlns="http://www.tei-c.org/ns/1.0" xmlns:xi="http://www.w3.org/2001/XInclude">
    <teiHeader>
        <fileDesc>
            <titleStmt>
                <title>Letter #6 from Robert Graves to William Graves (at Oundle School)
                        <date>November 15, 1957</date>
                </title>
            </titleStmt>
            <publicationStmt>
                <publisher>Digital Humanities at Oxford Summer School</publisher>
                <pubPlace>Oxford</pubPlace>
                <date when="2015-07">July 2015</date>
                <availability> [30 lines]
            </publicationStmt>
            <sourceDesc>
                <p>Letters from Robert and Beryl Graves to William Graves 1957-1967. 41 ALS and TLS,
                    56pp. , St John's College (Oxford) Library, Special Collections (Holmes 4) , Box
                    4 Gp17 </p>
                <p>ST JOHN'S COLLEGE LIBRARY COLLECTION OWNED LETTERS FROM ROBERT (1895-1985) &amp;amp;
                    BERYL (1915-2003) GRAVES TO WILLIAM GRAVES (1940- ) </p>
            </sourceDesc>
        </fileDesc>
        <profileDesc>
            <textClass>
                <catRef scheme="#genre" target="#correspondence"/>
            </textClass>
            <langUsage>
                <language ident="en">English</language>
            </langUsage>
            <particDesc> [240 lines]
            <settingDesc> [299 lines]
            <correspDesc> [10 lines]
        </profileDesc>
    </teiHeader>
```

TEI/XML document viewed in text mode



TEI/XML document viewed in a tree editor

Testing XPath expressions

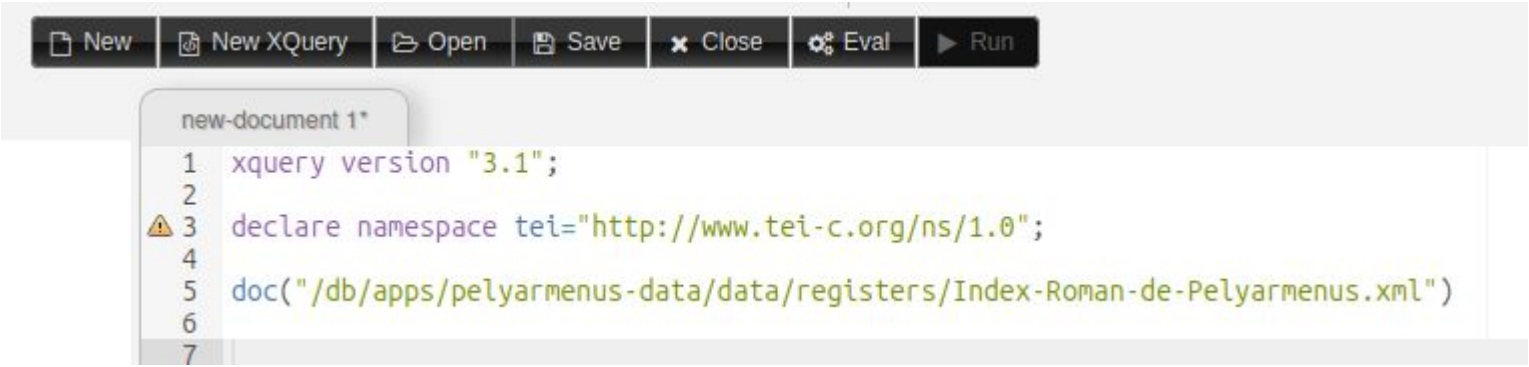
- **Editor**: Go to http://localhost:8080/exist/eXide/index.html
- **Namespace**:
  - To query TEI documents, you need to declare the TEI namespace

```
declare namespace tei="http://www.tei-c.org/ns/1.0";
```

Testing XPath expressions

- **Context**:
  - specify the context with **`doc()`** or **`collection()`** functions, otherwise the entire database is searched

`doc("/db/apps/pelyarmenus-data/data/registers/Index-Roman-de-Pelyarmenus.xml")`



Testing XPath expressions

- Get all person names (persName element) of the document
- After writing the expression, click on the **Eval** button

Testing XPath expressions

- Get all person names (persName element) of the document
- After writing the expression, click on the **Eval** button

```
doc("/db/apps/pelyarmenus-data/data/registers/Index-Roman-de
-Pelyarmenus.xml")//tei:persName
```

Testing XPath expressions

1. Get the **<persName>** elements that are a direct child of **<person>**
2. Get the **<persName>** elements that have a **@ref** attribute
3. Count how many **<persName>** elements have a **@ref** attribute and
4. How many **<persName>** elements do not have a **@ref** attribute
5. Get the **<persName>** elements whose **@ref** attribute is equal to **“#Edypus2”**
6. Get the **<persName>** element whose **@ref** attribute starts with **“#Edypus”**

TEI Processing model

TEI Processing Model

- The keystone of TEI Publisher: the implementation of the TEI Processing Model
- It formalizes the description of transformation rules in TEI (implementation independent)
- Officially part of the TEI Guidelines since 2016
- More sustainable and usually faster than customized, project-specific transformations
- It is the source of all TEI Publisher textual transformations

TEI Processing Model

How editors and developers communicate?

"this is a block", "this is a footnote", "a note on the margin", "an inline fragment set out in italic and highlighted", "fragment marked in <>", "a first-level heading"

Note this quite limited vocabulary - gave rise to the concept of "**behaviours**"

TEI Source

```
<TEI>
  <text>
    <body>
```

TEI Fragment

```
<div type="chapter">
  <head>
    <p>
```

```
<div type="chapter">
  <head>
    <list>
      <item>
```

ODD

```
<elementSpec ident="div">
  <model predicate="@type='chapter'"
    behaviour="section">
  <model behaviour="block">

<elementSpec ident="head">
  <model behaviour="heading">

<elementSpec ident="list">
  <model behaviour="list">
```

HTML FO

LaTeX ePub

Customization process

- Write the TEI ODD XML by hand
- Easier and safer: using the ODD editor (accessible through a URL like: https://workshop.jinntec.de/exist/apps/tei-publisher/odd-editor.html)

TEI representation

elementSpec

modelSequence

model

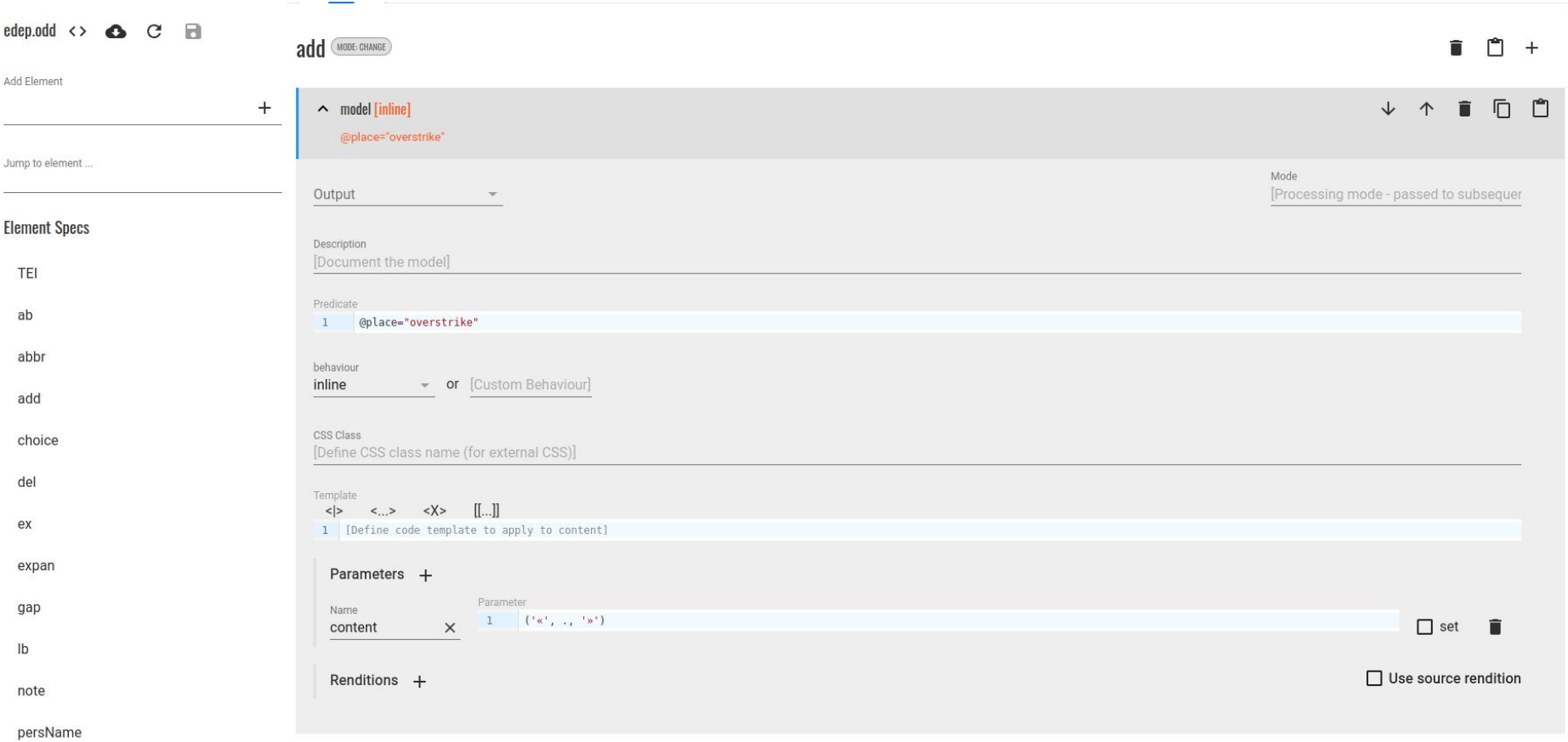
@predicate

@behaviour

@cssClass

outputRendition

```
<elementSpec ident="name" mode="change">
    <model predicate="@type='place'" behaviour="inline">
        <outputRendition>
            font-variant: small-caps;
        </outputRendition>
    </model>
    <model behaviour="inline">
        <outputRendition>
            color: #CC6600;
        </outputRendition>
    </model>
</elementSpec>
```


edep.odd
Add Element
Jump to element ...
Element Specs
TEI
ab
abbr
add
choice
del
ex
expan
gap
lb
note
persName
add
MODE: CHANGE
model [inline]
@place="overstrike"
Output
Mode
[Processing mode - passed to subsequer
Description
[Document the model]
Predicate
1 @place="overstrike"
behaviour
inline
or
[Custom Behaviour]
CSS Class
[Define CSS class name (for external CSS)]
Template
<|> <...> <X> [[...]]
1 [Define code template to apply to content]
Parameters
Name
content
Parameter
1 ('«', ., '»')
set
Renditions
Use source rendition

Behaviours library

- **alternate**
- anchor
- **block**
- body
- break
- cell
- cit
- document
- figure
- graphic
- heading
- **inline**
- list
- listItem
- metadata
- note
- **omit**
- paragraph
- **pass-through**
- row
- section
- table
- text
- title
- webcomponent

Predicate

- In the `@predicate` attribute we establish the condition that must be met for that model to be applied
- The value of `@predicate` is an XPath expression

XPath: expression language for XML

Applying models

Only the **first** matching model will be applied. **Simple and predictable!**

Matching means without any predicate or with a predicate for which its condition evaluates to true

Nota bene: Any following models will be **ignored** even if they match

How is it going to look like?
cssClass vs outputRendition

Normally you should assign a CSS class through the `cssClass` attribute to be used in external CSS stylesheets

`outputRendition` recommended only for strictly editorial aspects, e.g. convention to add [ ] to mark conjectures

general appearance (like margins, typographical elements etc) should be controlled by the external CSS theme for the application

Model parameters

- All behaviors take at least one implicit parameter which corresponds to the content to be processed
  - If the parameter `content` is not specified, the current node will be processed
- Some behaviours need additional parameters, e.g.:
  - link: `uri`
- Alternate behaviour has a different name for the content to be processed:
  - alternate: `default` and `alternate`

Model parameters

```
<elementSpec ident="date" mode="change">
    <model output="web" predicate="@when" behaviour="alternate">
        <param name="default" value="."/>
        <param name="alternate" value="format-date(@when, '[FNn], [D1o] [MNn], [Y]')"/>
    </model>
    <model predicate="text()" behaviour="inline"/>
</elementSpec>
```

Control and adjust the transformation process through parameters.

modelSequence

```
<elementSpec ident="title" mode="change">
    <modelSequence predicate="parent::titleStmt/parent::fileDesc">
        <model predicate="preceding-sibling::title" behaviour="text">
            <param name="content" value="' - '"/>
        </model>
        <model behaviour="inline"/>
    </modelSequence>
</elementSpec>
```

For cases when we want more than one model to be applied, e.g. to display the verse number and the verse content

Hands on session

Check the assignment on handling the apparatus