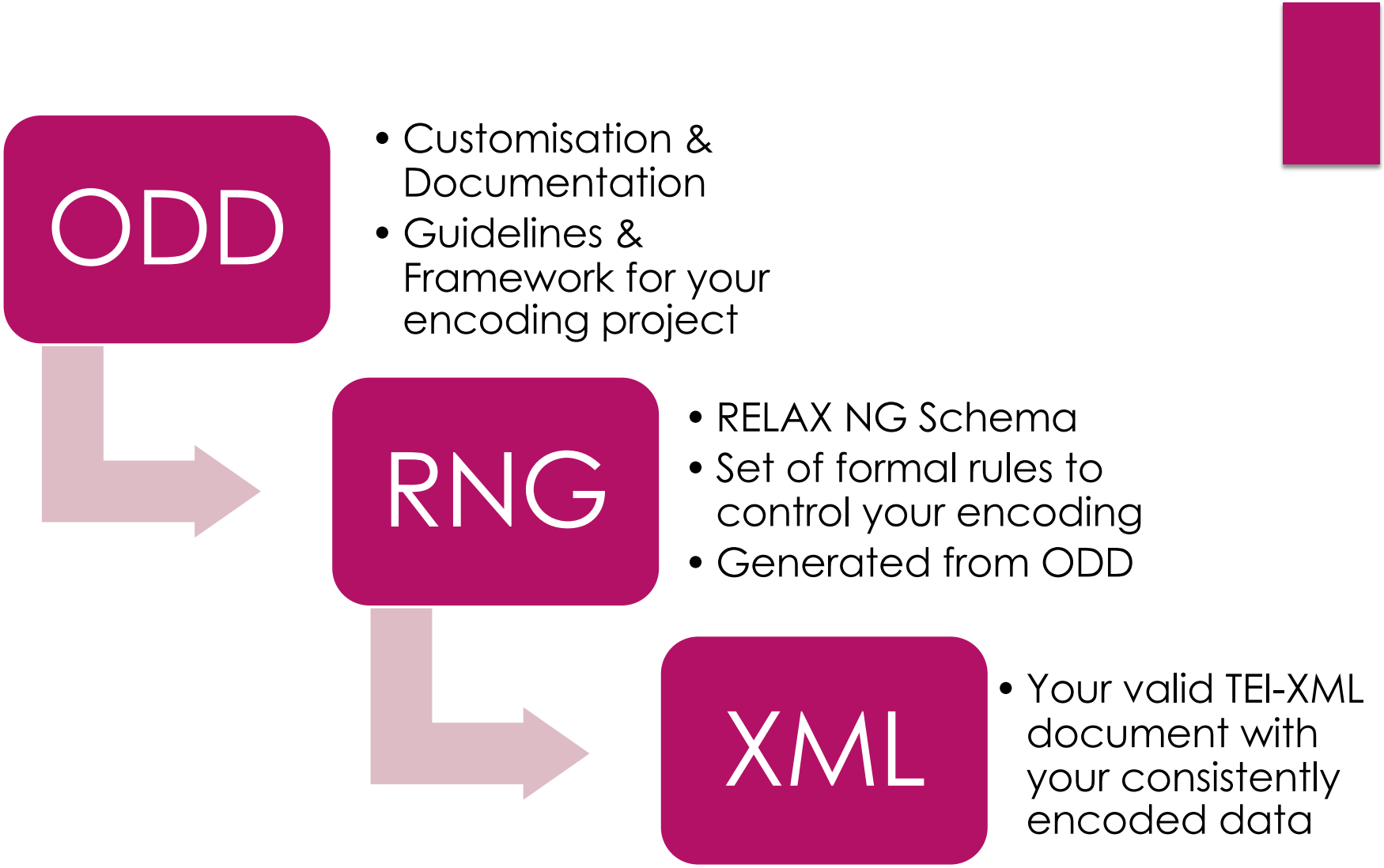
Modelling humanities data with TEI-XML

SCHOLARLY EDITING AND MANUSCRIPT CATALOGUING IN THE DIGITAL AGE

Dr Katarzyna Anna Kapitan
11 December 2024

Recap: ODD

```
<TEI xml:lang="en">
  <teiHeader> Metadata </teiHeader>
  <text>
    <body>
      <p> Description of the schema </p>
      <schemaSpec ident="test" targetLang="en"> […]
        <moduleRef key="module_name"/>
        <elementSpec ident="element_name" mode="mode_of_change"
    module="module_name">
        […]
        </elementSpec></schemaSpec></body></text></TEI>
```


ODD
• Customisation & Documentation
• Guidelines & Framework for your encoding project
RNG
• RELAX NG Schema
• Set of formal rules to control your encoding
• Generated from ODD
XML
• Your valid TEI-XML document with your consistently encoded data

Exercise 1.1: Autogenerated ODD from XML

- Create an ODD file from **minimal_encoding.xml** by using the **oddbyexample.xsl.**
  - Tutorial: Burnard_2013_How_to_Make_an_ODD_Automagically.pdf
- Generate **RELAX NG** schema and **XHTML** guidelines from your ODD.
- Associate the RNG schema with with your XML file (**minimal_encoding.xml**),
  - Does it validate correctly?
  - What if you tag "Paris" with **placeName**?

Exercise 1.2: Autogenerated ODD from XML

- Edit the ODD file so that it additionally allows the **placeName** element in the main body of the text.
- Document this change by providing a brief **description** and one usage **example**.
- Edit the ODD file so that it additionally allows the **placeName** element in the main body of the text.
- Document this change by providing a brief **description** and one usage **example**.

Schematron

Recap: Schema Languages

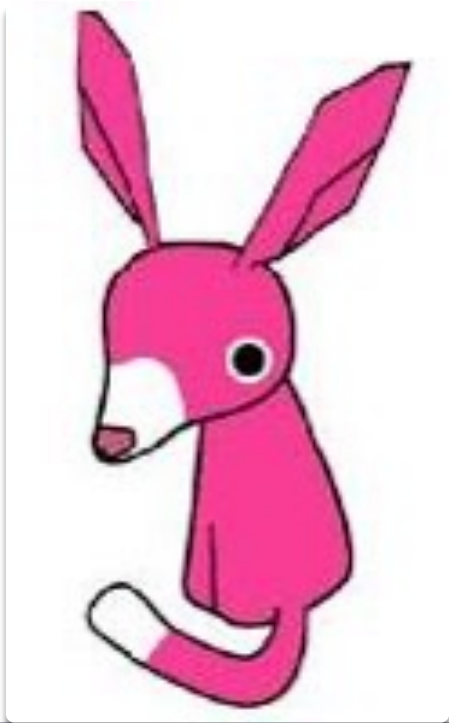
- Source: Syd Bauman, « A TEI customization for writing TEI customizations », Journal of the Text Encoding Initiative [Online], Issue 12 | July 2019 -, Online since 15 November 2019. URL: http://journals.openedition.org/jtei/2573

Table 1. Some of the schema languages for XML documents, arranged roughly by family of language.

SGML Document Type Declaration Family	W3C Schema Language Family	Regular Expression Family	Others
DTD	XML-Data	RELAX	DSD
XDTD	XDR	XDUCE	*Schematron*
DTD++	DCD	TREX	Examplatron
DTD++ 2.0	SOX	*RELAX NG*	X-definition
	DDML		*TEI ODD*
	XSD		

Schematron

- Is a formal schema language to express rules for XML documents.
- Is a relatively simple but powerful validation language
- Is expressed in XML
- Relies heavily on **XPath**, which means that if you want to write **Schematron** schemas, you have to understand **XPath** (Week 8)



- Source: Erik Siegel, *Schematron: A language for validating XML* (2022)

Schematron

- In Schematron you define all the **error** and **report** messages in your own words!
- There are two types of Schematron rules:
  - **Assertions:** when the condition for an assertion fails, an error message is issued.
  - **Reports:** when the condition for a report hold, a report message is issued.

Assertion

<sch:assert test="*@type*">

The type attribute is required.

</sch:assert>

- Make sure that the type attribute (@type) is present
- When there is **NO** @type, give me an error

Report

```
<sch:report test="not(child::surname)" role="error">
    The surname element is required.
</sch:report>
```

- Make sure that there is **no surname** element as a child of my context node and give me an error message when this is the case.

Schematron Rules

In **Schematron**, the context item is set using **@context** attribute of the **rule** element.

```
<sch:rule context="author" >
        <sch:assert test="child::surname" role="error">
                Each author entry needs a surname
        </sch:assert>
</sch:rule>
```

Schematron rules & XPath

```
<sch:rule context="tei:lg[@type='limerick']">
        <sch:report test="count( child::tei:l ) != 5" role="error">
            A limerick should not have
            <sch:value-of select="count( child::tei:l )"/>
            lines.
        </sch:report>
</sch:rule>
```

Assertion or Report

Assertion:

```
<sch:assert test="count( child::tei:l ) = 5" role="error">
A limerick should  have 5 lines
</sch:assert>
```

Report:

```
<sch:report test="count( child::tei:l ) != 5" role="error">
    A limerick shouldn't have <sch:value-of select="count( child::tei:l )"/> lines.
</sch:report>
```

Stand-alone Schematron & name spaces

- <?xml version="1.0" encoding="UTF-8"?>
  <sch:schema **xmlns:sch="http://purl.oclc.org/dsdl/schematron"**
  queryBinding="xslt2" xmlns:sqf="http://www.schematron-quickfix.com/validator/process">
  <sch:ns uri="http://www.tei-c.org/ns/1.0" prefix="tei"/>
  <sch:pattern>
  <**sch:**rule context="**tei:**lg[*@type*='limerick']">
  <**sch:**report test="*count*(child::**tei:l**) != 5" role="error">A limerick should not
  have <sch:value-of select="*count*(child::**tei:l**)"/> lines</sch:report>
  </sch:rule></sch:pattern></sch:schema>

Exercise 2: Stand-alone Schematron

- Open **test_poem_tei.xml** and **test_poem_schematron.sch**
- Associate the SCH schema with your XML, validate it.
- Fix the errors in XML, so it validates correctly.
- In SCH file make the **rhyme** attribute required for lg
- In XML delete the rhyme attribute from lg, check whether your file validates correctly.

Schematron in ODD

- You can use schmatron rules within **elementSpec**
- Two new elements **constraintSpec** & **constraint**:
- <constraintSpec ident="*identifier*" mode="*mode_of_change*" scheme="**isoschematron**">

  <constraint> Here goes your Schematron rule </constraint>
  </constraintSpec>

```
<elementSpec ident="lg" mode="change">
   <constraintSpec ident="lg_limerick" mode="change"
    scheme="isoschematron">
          <constraint>
             <sch:rule context=".[@type='limerick']">
                <sch:report test="count( child::tei:l ) != 5" role="error">
                   A limerick should not have
                   <sch:value-of select="count( child::tei:l )"/> lines.
                   It should have 5 lines. Fix your encoding.
                </sch:report>
             </sch:rule>
          </constraint>
   </constraintSpec>
</elementSpec>
```

Exercise 3: Schematron in ODD

- Using **oddbyexample.xsl** (Ex1) & **test_poem_tei.xml** (Ex 2) create an ODD file.
- Edit the ODD file to incorporate the following schematron rule for **line group** with **@type 'limerick'**

```
<sch:report test="count( child::tei:l ) != 5" role="error">
```

- Export your ODD into RELAX NG.
- Validate **test_poem_tei.xml** with your new RELAX NG.

More about Schematron

- Erik Siegel, *Schematron: A language for validating XML* (2022)