# Dos and don’ts in TEI schema customization

…

## An introduction to the ODD specification

**Helena Bermúdez-Sabel**
software developer at Jinntec

# Outline

- Preliminary notes
- Rationale: interoperability and reusability with TEI
- TEI conformance and ODD
- Good practices when writing TEI specifications
- Resources

# Prelude

# TEI ODD

One Document Does it All

> a TEI XML format used to express schema fragments, prose documentation, and reference documentation for any XML markup scheme as a single document. It is the language in which the TEI is itself defined, and which should also be used to express a customization of the TEI scheme.

[Source]

```
<schemaSpec ident="correspondence-schema" start="TEI">
   <moduleRef key="header"/>
   <moduleRef key="core" except="gb meassure meassureGrp postBox rb rt ruby stage handNote "/>
   <moduleRef key="tei" except="div1 div2 div3 div4 div5 div6 div7"/>
   <moduleRef key="textstructure"/>
   <moduleRef key="msdescription"/>
   <moduleRef key="figures" except="notatedMusic"/>
   <moduleRef key="analysis" except="cl interp interpGrp m phr span spanGrp"/>
   <moduleRef key="linking" except="alt altGrp joinGrp timeline when"/>
   <moduleRef key="namesdates" except="climate terrain offset"/>
   <moduleRef key="textcrit"/>
   <moduleRef key="transcr"
      except="fw listTranspose mod path secl surface surfaceGrp transpose zone"/>
   <elementSpec ident="profileDesc" mode="change" module="header">
      <desc>provides a detailed description of non-bibliographic aspects of a text, specifically the languages and
sublanguages used, the situation in which it was produced, the participants and their setting. In this project, it is
mandatory to include here a <gi>correspDesc</gi> element to describe the actions related to the
correspondence</desc>
      <content>
         <elementRef key="correspDesc"/>
         <classRef key="model.profileDescPart" minOccurs="0" maxOccurs="unbounded"/>
      </content>
   </elementSpec>
<!-- it continues  -->
</schemaSpec>
```

# Rationale

# No interoperability nor reusability without a *good* schema

208

# 208

Number of elements that are possible children of <p> (P5 4.7.0)

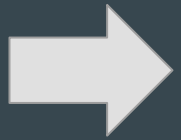
# 586 & 270

Number of elements & attributes TEI P5 Version 4.7.0.

# The importance of TEI customizations

Documentation and schema specifications are necessary:

- to control the consistency of the encoding
- to document and describe the data model (which corresponds to the real informational requirements of the project)



Usability

Exploitation through existing tools

Reusability

Interoperability

# Conformance and validation

# What is TEI conformance

# What is TEI conformance

## ...besides a promise added in grant applications

# TEI conformance

A document is *TEI-conformant* if it:

- is a well-formed XML document
- can be validated against a TEI Schema
- conforms to the TEI Abstract Model
- uses the TEI namespace (and other namespaces where relevant) correctly
- is documented by means of a TEI-conformant ODD file which refers to the TEI Guidelines

[Source]

# TEI conformance

A document is *TEI-conformant* if it:

- is a well-formed XML document
- can be validated against a TEI Schema
- conforms to the TEI Abstract Model
- uses the TEI namespace (and other namespaces where relevant) correctly
- **is documented by means of a TEI-conformant ODD file which refers to the TEI Guidelines**

[Source]

# TEI validation constraint

> All TEI-conformant documents must validate against a schema file that has been derived from the published TEI Guidelines, combined and documented in the manner described in section 23.3 Customization. We call the formal output of this process a TEI Schema.
>
> [Source]

# Customizations provided by the TEI-C and TEI community

[See here for a longer list]

- Lite: A slimmed-down version of TEI intended “to meet 90% of the needs of 90% of the TEI user community”
- TEI simplePrint: An entry-level customization, focused primarily on the needs of those encoding Western European early modern printed material
- Best Practices for TEI in Libraries: A guide for mass digitization, automated workflows, and promotion of interoperability with XML using the TEI
- TEI Lex-0: A technical specification and a set of community-based recommendations for encoding machine-readable dictionaries.

# Good practices

# Targeted audience

- Project members
  - Researchers
    - Documenting decisions
    - Keeping a record of the changes
  - Encoders
    - Guidelines with references for encoding
  - Developers
    - Description of the data model
- Scientific community
  - Those interested in creating interoperable datasets
  - Those interested in using the dataset

# Customization as part of the data

It guarantees that the value of the data is maintained over time, assuring its reusability

- Share the ODD as part of the TEI dataset
- The ODD should document your current encoding practice
- Associate your ODD to a specific release of the TEI (see attribute `@source` in the `<schemaSpec>` element)

# ODD chaining

Possible workflows:

- Select as source ODD a TEI subset (see TEI Lex-0, Epidoc) or the ODD of a related project
- Within the same project, use chaining, e.g., for separation of concerns, increase interoperability (see ELTeC, Humboldt Digital, Beta masāhəft)

# Check pointers

Use Schematron within ODD to check that the URIs in your documents:

- point to something that can be retrieved
- point to the expected type of object

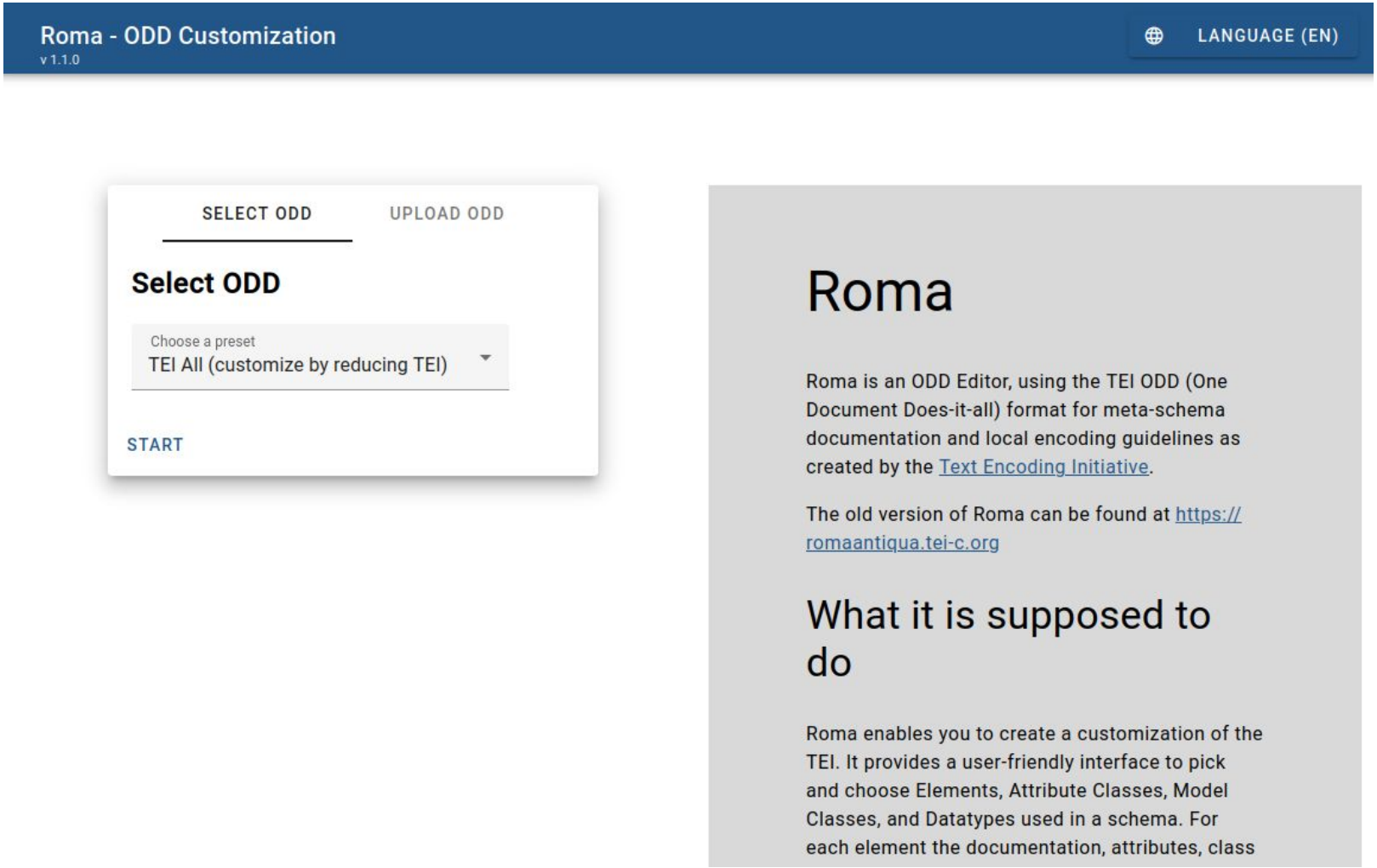
```
<constraintSpec ident="g_points_to_local_char_or_glyph" scheme="schematron">
  <desc>Example of how to test that the <att>ref</att> of <gi>g</gi> points to a local
      <gi>char</gi> or <gi>glyph</gi>.</desc>
  <constraint>
    <sch:rule context="tei:g/@ref">
      <sch:let name="ref" value="substring( normalize-space(.), 2 )"/>
      <sch:assert test="id( $ref )/self::tei:char | id( $ref )/self::tei:glyph"> The ref= of
        'g' is supposed to point to a 'char' or 'glyph'; this one ("<sch:value-of select="."
        />") points to a '<sch:value-of select="local-name( id( $ref ) )"/>'. </sch:assert>
    </sch:rule>
  </constraint>
</constraintSpec>
```

## Customizing TEI to Check Pointers

# Resources

# Tools for ODD processing

- Roma: a web-based service that provides a user-friendly interface for creating ODD customization files and generating schemas and documentation from them.
- roma: a command-line tool for generating schemas and documentation from ODD customizations.
- TEIGarage: a web service which offers various transformations to and from TEI.
- command-line scripts that are part of the TEI Stylesheets package: see source code.
- oXygen XML Editor version 13.1 or later. See also the oXygen plugin maintained by the TEI-C.


Roma - ODD Customization
v 1.1.0
LANGUAGE (EN)
SELECT ODD
UPLOAD ODD
Select ODD
Choose a preset
TEI All (customize by reducing TEI)
START
Roma
Roma is an ODD Editor, using the TEI ODD (One Document Does-it-all) format for meta-schema documentation and local encoding guidelines as created by the Text Encoding Initiative.
The old version of Roma can be found at https://romaantiqua.tei-c.org
What it is supposed to do
Roma enables you to create a customization of the TEI. It provides a user-friendly interface to pick and choose Elements, Attribute Classes, Model Classes, and Datatypes used in a schema. For each element the documentation, attributes, class

```
Usage: roma [options] schemaspec [output_directory]
options, shown with defaults:
–xsl=/usr/share/xml/tei/stylesheet
–teiserver=http://www.tei-c.org/Query/
–localsource= # local copy of P5 sources options, binary switches:
–doc # create expanded documented ODD (TEI Lite XML)
–lang=LANG # language for names of attrbutes and elements
–doclang=LANG # language for documentation
–dochtml # create HTML version of doc
–patternprefix=STRING # prefix relax patterns with STRING
–docpdf # create PDF version of doc
–nodtd # suppress DTD creation
–norelax # suppress RELAX NG creation
–noxsd # suppress W3C XML Schema creation
–noteic # suppress TEI-specific features
–debug # leave temporary files, etc.
```

## *TEIGarage Conversion*

**Please select the type of the document you want to convert**

**Convert from:** ?

**Documents**

**Presentations**

**Spreadsheets**

# Stylesheets


TEI XSL Stylesheets

This is a family of XSLT 3.0 stylesheets to transform TEI XML documents to various formats, including XHTML, LaTeX, XSL Formatting Objects, ePub, plain text, RDF, JSON; and to/from Word OOXML (docx) and OpenOffice (odt). They concentrate on the core TEI modules which are used for simple transcription and "born digital" writing. It is important to understand that they do *not*:

- cover all TEI elements and possible attribute values
- attempt to define a standard TEI processing or rendering model

and should not be treated as the definitive view of the TEI Consortium.

For more information, see https://tei-c.org/tools/stylesheets/

```
$ teitohtml --help

TEI conversion: from tei to html

  Usage: /usr/bin/teitohtml [options] inputfile [outputfile]

  Options, binary switches:
  --verbose            # be verbose
  --debug              # be verbose, do not delete intermediate files
  --apphome=/usr/share/xml/tei/stylesheet   # where to find app directory
  --profiledir=/usr/share/xml/tei/stylesheet/profiles    # where to find profile directory
  --profile=default    # which transformation profile to use
  --oxygenlib=/usr/share/oxygen/lib    # where is oxygenlib
  --odd                # perform processing of ODD (if appropriate)
  --localsource=DIR    # where is local copy of source of TEI Guidelines
  --summaryDoc         # only make summary, when doing ODD processing

  Options, shown with defaults:
  --saxonjar=/usr/share/saxon/saxon10he.jar  # location of Saxon jar file
```

# oXygen XML Editor frameworks for TEI

This project contains the oXygen XML Editor specific support for TEI. It is organized as follows:

- the "frameworks" folder contains the actual TEI frameworks.
- the "lib" folder contains the jars required for building the JAVA extensions from each framework.
- the "tools" folder is intended for required tools like the Apache ANT.
- the "dist" folder will contain the packed distribution of all frameworks.

https://github.com/TEIC/oxygen-tei

# atop

Another TEI ODD Processor

This is a pre-release partially-complete version of the ATOP processor. See the documentation in Documentation/end-user to find out what it can currently do and how you might use it.

https://github.com/TEIC/atop

# ODD updates and Continuous Integration

Some examples:

- Lyon in Mourning Project Schema
- Beta masāhəft: Manuscripts of Ethiopia and Eritrea

# Tutorials

- From the Women Writers Project resources for teaching and learning TEI, see *TEI Customization Primer*
- TEI Council: *Getting Started with P5 ODDs*
- Terras, Melissa, Edward Vanhoutte, and Ron Van den Branden. Module 8: Customising TEI, ODD, Roma. TEI by Example
- Bournard, Lou. 2016. ODD chaining for Beginners
- Bauman, Syd. 2022. *Customizing TEI to Check Pointers*

# Bibliography


- Bauman, Syd. 2019. A TEI Customization for Writing TEI Customizations. *Journal of the Text Encoding Initiative*, 12. doi:10.4000/jtei.2573.
- Burnard, Lou, Christof Schöch and Carolin Odebrecht. 2021. In search of comity: TEI for distant reading", *Journal of the Text Encoding Initiative*, 14. doi:10.4000/jtei.3500
- Burnard, Lou, and Sebastian Rahtz. 2013. A Complete Schema Definition Language for the Text Encoding Initiative. In *XML London 2013: Conference Proceedings*, 152–61. London: XML London. doi:10.14337/XMLLondon13.Rahtz01.
- Rahtz, Sebastian, and Lou Burnard. 2013. Reviewing the TEI ODD System. *DocEng '13: Proceedings of the 2013 ACM Symposium on Document Engineering*, 193–96. New York: ACM. doi:10.1145/2494266.2494321.

# Thank you!

helena.bermudez@jinntec.de