

# TEI to CRM

Sebastian Rahtz

TEI members meeting, Zadar, 2010-11-13



## Task

...all based on a chimerical and simplistic idea that you can take an arbitrary TEI document which models everything you know in a single XML vocabulary, and then extract useful RDF assertions in CIDOC CRM for **interchange**.

- record the relationship of TEI elements to known CIDOC CRM concepts in a formal way, maintained in a single document with the mapping guidelines
- provide actual mapping code to get from TEI XML to RDF XML
- consider workflow and recommendations on tailoring conversion
- target is a module for OxGarage — upload your TEI XML file, get back an RDF file

Contrast research on extracting assertions from published literature using NLP (eg Stellar/Star work on archaeological grey literature).



## Background to my work

CLAROS project based at Oxford aims to combine discrete databases of information about the ancient world using an RDF triplestore of assertions using CIDOC CRM.

Currently includes art objects, archaeological sites, antiquarian photographs, and onomastics.

Lexicon of Greek Personal Names contributes via representation in TEI XML

## Basic tool for recording mapping: <equiv>

```
<elementSpec ident="event" mode="change">
  <equiv
    filter="crm.xsl"
    mimeType="text/xsl"
    name="E5"
    uri="http://erlangen-crm.org/101001/E5_Event"/>
</elementSpec>
```

**name** names the underlying concept of which the parent is a representation

**uri** references the underlying concept of which the parent is a representation by means of some external identifier

**filter** references an external script which contains a method to transform instances of this element to canonical TEI

**mimeType** MIME media type of filter script

## Workflow

Read ODD, extract filter info from each `<equiv>` and use it to generate a wrapper XSLT script

```
<XSL:stylesheet version="2.0"
  xpath-default-namespace="http://www.tei-c.org/ns/1.0">
  <XSL:import href="crm.xsl"/>
  <XSL:template match="*[ancestor::teiHeader]"/>
  <XSL:template match="*">
    <XSL:apply-templates
      select="*|@*|processing-instruction()|comment()|text()"/>
  </XSL:template>
  <XSL:template
    match="text()|comment()|@*|processing-instruction()"/>
  <XSL:template match="person">
    <XSL:call-template name="E21"/>
  </XSL:template>
  <XSL:template match="place">
    <XSL:call-template name="E53"/>
  </XSL:template>
  <XSL:template match="persName">
    <XSL:call-template name="E82"/>
  </XSL:template>
  <XSL:template match="placeName">
    <XSL:call-template name="E48"/>
  </XSL:template>
  <XSL:template match="event">
```

# What is in crm.xsl?

```
<xsl:template name="E47">
  <P87_is_identified_by>
    <E47_Place_Spatial_Coordinates>
      <value>
        <xsl:value-of select="."/>
      </value>
    </E47_Place_Spatial_Coordinates>
  </P87_is_identified_by>
</xsl:template>
<xsl:template name="E69">
  <P100i_died_in>
    <E69_Death>
      <P4_has_time-span>
        <E52_Time-Span>
          <P82_at_some_time_within>
            <E61_Time_Primitive>
              <xsl:call-template name="calc-date-value"/>
            </E61_Time_Primitive>
          </P82_at_some_time_within>
        </E52_Time-Span>
      </P4_has_time-span>
    </E69_Death>
  </P100i_died_in>
</xsl:template>
```

# Input

```
<person xml:id="ArnMag01" sex="1" role="scholar">
  <persName xml:lang="is">Árni Magnússon</persName>
  <persName xml:lang="la">Arnas Magnæus</persName>
  <persName xml:lang="da">Arne Magnusson</persName>
  <birth when="1663-11-13">13 November 1663</birth>
  <death when="1730-01-07">7 January 1730</death>
  <residence>
    <date from="1663" to="1680">1663-1680</date>
    <placeName>
      <settlement type="farm">Hvammur</settlement>
      <region type="county">Dalasýsla</region>
      <region type="compass">Western</region>
      <country key="IS">Iceland</country>
    </placeName>
  </residence>
  <residence>
    <date from="1680" to="1683">1680-1683</date>
    <placeName>
      <settlement type="institution">Skálholt</settlement>
      <region type="county">Árnessýsla</region>
      <region type="compass">Southern</region>
      <country key="IS">Iceland</country>
    </placeName>
  </residence>
  <residence>
    <date from="1683" to="1685">1683-1685</date>
```

# Result

```
<RDF>
  <E21_Person
    rdf:about="http://www.example.com/idArnMag01">
    <P131_is_identified_by xml:lang="is">
      <E82_Actor_Appellation>
        <value>Árni Magnússon</value>
      </E82_Actor_Appellation>
    </P131_is_identified_by>
    <P131_is_identified_by xml:lang="la">
      <E82_Actor_Appellation>
        <value>Arnas Magnæus</value>
      </E82_Actor_Appellation>
    </P131_is_identified_by>
    <P131_is_identified_by xml:lang="da">
      <E82_Actor_Appellation>
        <value>Arne Magnusson</value>
      </E82_Actor_Appellation>
    </P131_is_identified_by>
    <P98i_was_born>
      <E67_Birth>
        <P4_has_time-span>
          <E52_Time-Span>
            <P82_at_some_time_within>
              <E61_Time_Primitive>
                <value>1663-11-13</value>
              </E61_Time_Primitive>
            </P82_at_some_time_within>
          </E52_Time-Span>
        </P4_has_time-span>
      </E67_Birth>
    </P98i_was_born>
  </E21_Person>
</RDF>
```



# Problem of ambiguity of context

## Contrast

```
<place>  
  <placeName>Zadar</placeName>  
</place>
```

## with

```
<p>He was born in <placeName>Zadar</placeName>  
</p>
```

## Problem of ambiguity: <name>

```
<name type="place">Zadar</name>  
<name type="person">Oyvind</name>  
<rs type="person">Piotr</rs>
```

## The usual problems

- how to record location in TEI text of source claim
- date of claim
- how to actually express dates
- representing uncertainty and precision
- chronological periods
- how to actually express spatial coordinates

# Taxonomies, categories, @type etc

Do we

- understand how to use E55\_Type
- convert existing taxonomies and categories to SKOS notation, and refer to it
- enhance existing taxonomies with links to SKOS
- use informal @type from TEI documents to generate SKOS on the fly?
- something else?

## What more could we do with the ODD?

Use it to create a schema to check whether a document can be mapped

- remove elements which cannot be matched to CRM
- add constraints to check situations which stop meaningful mapping
- eg check whether a `<placeName>` has a `@ref`
- eg check whether a `<placeName>` has a `@xml:id`

## What next?

Can this simplistic approach work?

Todo:

- Complete mapping as it stands, with all info in <http://www.tei-c.org/SIG/Ontologies/guidelines/guidelinesTeiMappableCrm.xml> included
- Build up test suite
- Demonstrate whether the resulting RDF can be used for anything