

Future of CMDI: topic areas

- CMDI 1.2
 - Application of CMDI 'as it is'
- CMDI 1.3
 - Small(ish) backwards compatible changes
- CMDI 2.0
 - Fundamental changes to implementation, representation etc
- CMDI in CLARIN and beyond
 - CMDI 'governance'
 - Legal issues
 - CMDI & FAIR
 - Interoperability
 - CMDI outside CLARIN

tinyurl.com/cmd-future

Topics: CMDI 1.2

Scope: application of CMDI 'as it is'

- Application of “cues for tools”
 - Display, rendering cues etc
- Automatic value calculation/derivation

CMDI 1.2 – Cues for tools

- *DisplayPriority*: a number that is used in determining the *display value* for the containing component in clients and viewers;
- CLARIAH CMDI Forms (github.com/knaw-huc/clariah-cmdi-forms)
 - ***displayOrder***: can be used to change the order of elements and components, e.g., the mandatory and most used elements can be placed at the top of the form for less scrolling; it is even possible to mix elements and components, which is not possible in CMDI profiles;
 - ***hide***: hide the component, element or attribute; can be safely used with optional parts of the profile or should be combined with default or fallback auto values to prevent validation problems;
 - ***resource***: marks a component to which a resource (proxy) can be associated;
 - ***inputWidth***: set the width for an input field, e.g., to allow enough space to enter a long title;
 - ***inputHeight***: set the height for an input field, e.g., to allow enough space to enter a description of multiple lines of text.
 - ***resource***: resource proxies (of a specific type) can be associated with this component.

CMDI 1.2 – Auto value “functions”

- CLARIAH CMDI Forms (github.com/knaw-huc/clariah-cmdi-forms)
 - **now**: will set the value to the current date and/or time;
 - **fileSize**: will set the value to the file size of the associated file, i.e., the nearest resource associated with an ancestor component;
 - **fileModification**: will set the date and/or time of the associated file;
 - **default**: will set the default value for the field;
 - **fallback**: will set the value if the user does not enter any.

CMDI 1.2 – Remarks on cues and auto values

The CMDI 1.2 spec keeps them open, i.e., we can specify them ourselves

- Agree on a core set, e.g., *displayOrder* and *now*
- Have editor/viewer/institute specific prefixes, e.g., *ccf-hide*

Not used for

- Multilingual label, because where to specify the language?
 - `@cue:label-nl="Omvang" *`
 - see CMDI 1.3 - Multilinguality

Topics: CMDI 1.3(+?)

Scope: small(ish) backwards compatible changes

- Complete support for multilinguality
 - Labels in closed vocabularies
 - Labels for elements/components
- CCR: generic concepts
- Vocabularies
- Enhanced Interoperability
 - Tolerance towards arbitrary foreign name spaces
 - CMDI & linked data
 - What can be done without breaking backwards compatibility?

CMDI 1.3 - Multilinguality

- CMDI has now **partial multilingual support**:
 - **Documentation** in a profile/component can be in multiple languages;
 - **Free text elements** (value scheme *string*) can be repeated in multiple languages (multilingual is *true*);
 - **External (CLAVAS) vocabularies** might contain multilingual labels for vocabulary items (SKOS(-XL) labels).
- Missing:
 - **Multilingual labels** in the profile/component specification for components, elements, attributes and embedded vocabulary items.

```
<Element name="category">
  <Label xml:lang="nl">Categorie</Label>
  <ValueScheme>
    <Vocabulary>
      <enumeration>
        <item>
          <value>Dialectology/language variation</value>
          <label xml:lang="nl">Dialectologie/taalvariatie</label>
```


CMDI 1.3 – CCR: generic concepts

- The goal of the CCR coordinators is to recommend a (limited) set of generic concepts that can be combined to capture the semantics of a specific context:
 - Name
 - Identifier
 - Country
 - Language
 - Country + Name, Country + Identifier
 - Language + Name, Language + Identifier
- Will need the functionality of associating multiple concepts with a component/element/attribute/value
 - <Element name="iso-639-3-code" ConceptLink=".../CCR_C-3286_c2f2fcb2-26bd-389a-4fd0-a4e54111ccd4.../CCR_C-6193_096cfc32-04ce-c4d1-5ed4-5e92db53aa70" ...

CMDI 1.x – Vocabularies: CLAVAS

- Current:
 - ISO 639-3
 - Old
 - Versioning?
- Candidates:
 - **Resource types** by Metadata Curation TF
 - **Licenses** by CLARINO
 - **Media types** by IANA
 - + **format variants** by CLARIN(-D) (?)
- Prerequisites
 - Ownership should be clear
 - Metadata Curation TF as owner?
 - First consent by SCCTC ... stalled
- How to get moving? What should be our priorities?

CMDI 1.3 - Allow use of arbitrary foreign namespaces

- **Original situation (\leq CMDI 1.1):** no arbitrary foreign namespace content
- **Current situation (\geq CMDI 1.2):** <Resources> section of header is open for attributes
 - Main use case: add repository specific details (e.g. local path or identifier)
- **Mild proposal:** open up entire header
 - Use case: provenance (for example enrich record during harvest)
 - Use case: rights statements regarding record
- **Wild proposal:** open up entire payload section
 - Use case: blend in 3rd party information and enrichments
 - annotations, assertions, ...
 - Challenge: keeping grip on CMDI, preventing ad hoc practices/divergence

CMDI 1.3 – Interoperability: Linked Data

- When resources and metadata are linked data its easy to cross the boundary between them (multiple times):
 - Find an audio recording of a language with the sound ŋ (the velar nasal)
 - Use WALS to find the languages
 - Use CMD to find recordings of these languages
 - <http://cmdi2rdf.meertens.knaw.nl/cmd2rdf/how> (last query)
 - Current experiment:
 - Select a lexical item from a lexicon, i.e., Xhosa
 - Use CMD to find an annotated recording of Xhosa
 - Use the phonetic form from the lexical item to locate a matching annotation
 - Need: a LD representation of an EAF (or equivalent)
 - Return the matching multimedia fragment

Topics: CMDI 2.0

Scope: fundamental changes to implementation, representation etc

- Representation: beyond XML
 - CMD2RDF

CMDI 2.0 – CMD2RDF

- <http://cmdi2rdf.meertens.knaw.nl/cmd2rdf/>
- <https://doi.org/10.5334/bbi.8>
- Main problem:
 - Attributes on elements force the creation of an additional “artificial” layer

```
_:languages1    a      cmd2:Actor_Languages ;
  cmdm:contains      _:language1 .
_:language1 a      cmd2:Actor_Languages_Language ;
  cmdm:hasElementValue      "nld" ;
  cmdm:containsAttribute _:code1 .
_:code1      a      cmd2:Actor_Languages_Language_codeAttribute ;
  cmdm:hasAttributeValue      "639-3" .
```

Topics: CMDI in CLARIN and beyond

- Modelling governance
 - Complete freedom?
 - Guidance?
 - Tight control?
- -> Unified interoperability, quality and semantics
- CMDI & FAIR
 - Main challenges: Licencing, FAIR vocabularies (concept links)
- Legal issues
 - Metadata right statements
 - Resource rights and access
- Can (should?) CMDI play a role outside of CLARIN?

tinyurl.com/cmd-future

09:00 - 10:30 Future of CMDI

Agenda

1. [~09:05 - 09:15] Past en present of CMDI
2. [~09:15 - 09:30] Topic pitches and selection
3. [~09:30 - 10:00] **Discussion (break-out?)**
4. [~10:00 - 10:30] **Conclusions**

11:00 - 12:30 CMDI and metadata curation task forces meeting

13:30 - 15:00 Metadata curation hands-on session