CLARIN
Common Language Resources and Technology Infrastructure

CMDI Modelling Best Practices

Twan Goosen
CLARIN ERIC
twan@clarin.eu

CLARIN-NL CMDI Tutorial
Utrecht, 4 March 2014

Flexibility

CMDI offers a **very flexible model**

Large degree of **freedom to the modeller**

Advantage:

- Almost **anything** can be modelled

Disadvantage:

- A lot of **choices** have to be made
- The **options** may not always be clear

Three rules of CMDI modelling…

…you should consider following if you want
your metadata to live a healthy life
in the CMDI ecosystem

First rule of CMDI modelling:

Consider the instance level

The instance level

The instance level describes **concrete resources**

What will be described in one instance?

- One resource file (image/video/audio)
- A session (e.g. media file + annotations)
- A corpus (one or more parts)
- A service (tool or web service)

The instance level

What is the right **level of granularity**?

How ‘deep’ should the description go into the resource?

Annotation ≠ metadata. Metadata should be valid and stable for whole resource.

But the line can be blurred…

Granularity guidelines: http://www.clarin.eu/file/1790

The instance level

The instance level may describe **a hierarchy**

- Any CMDI file can point to any other CMDI file.
- Using a **collection profile** is recommended.
- CMDI is not a relational database system!

The instance level

Various ways of **producing** an instance

- **Manually**
  - Think about editing ‘ergonomics’
- **Generated from a database**
  - Follow logical structure of the database?
- **Transformed from another metadata format**
  - Should the CMDI version be structurally similar?

Second rule of CMDI modelling:

Reuse (if you can)

Consider using an **existing profile**

There are ~150 published profiles in the Component Registry for describing media files, corpora, services, collections, …

CLARIN provides **recommended profiles** for collections, different resource types, and different metadata sources

http://clarin.eu/faq/there-list-recommend-components-and-profiles

Consider reusing **existing components**

There are ~850 published components in the Component Registry for describing actors, locations, technical details, …

CLARIN provides **recommended components** for language codes, countries, MIME types, …
http://clarin.eu/faq/there-list-recommend-components-and-profiles

Reusing existing components: **which?**

- Look at **name and description**
- Look at **recommendation** list (CMDI FAQ)
  - National content coordinator will evaluate and provide more recommendations
- Look at **group** or **creator**
  - Colleagues or trusted centre/project

Reusing existing components: **how?**

Found a **perfect match**?

- Make it part of your component or profile **by reference**

Needs **tweaking**?

- **Edit as new** and make amendments. Constraints and semantics will be copied.
- Works for profiles, too

Make your output reusable

If your component or profile is complete, **publish** it so that others can use it.

Whenever possible, **design for reusability**.

- Modular
- As generic as possible
- Good documentation
- Explicit semantics

Third rule of CMDI modelling:

Define the semantics

Defining semantics

Semantics are defined by assigning **data categories** to components, elements and attributes

Exploitation software (e.g. VLO) works with arbitrary CMDI profiles because it picks up the semantic markings in components.

What data category to use?

- Only use **ISOcat URI's** as 'concept links'
- Most tools have **special support** for ISOcat
- **Search function** of Component Registry provides appropriate selection
- Carefully **read description**
- Consider creating a **data category** yourself
  - Remember the do's and dont's!

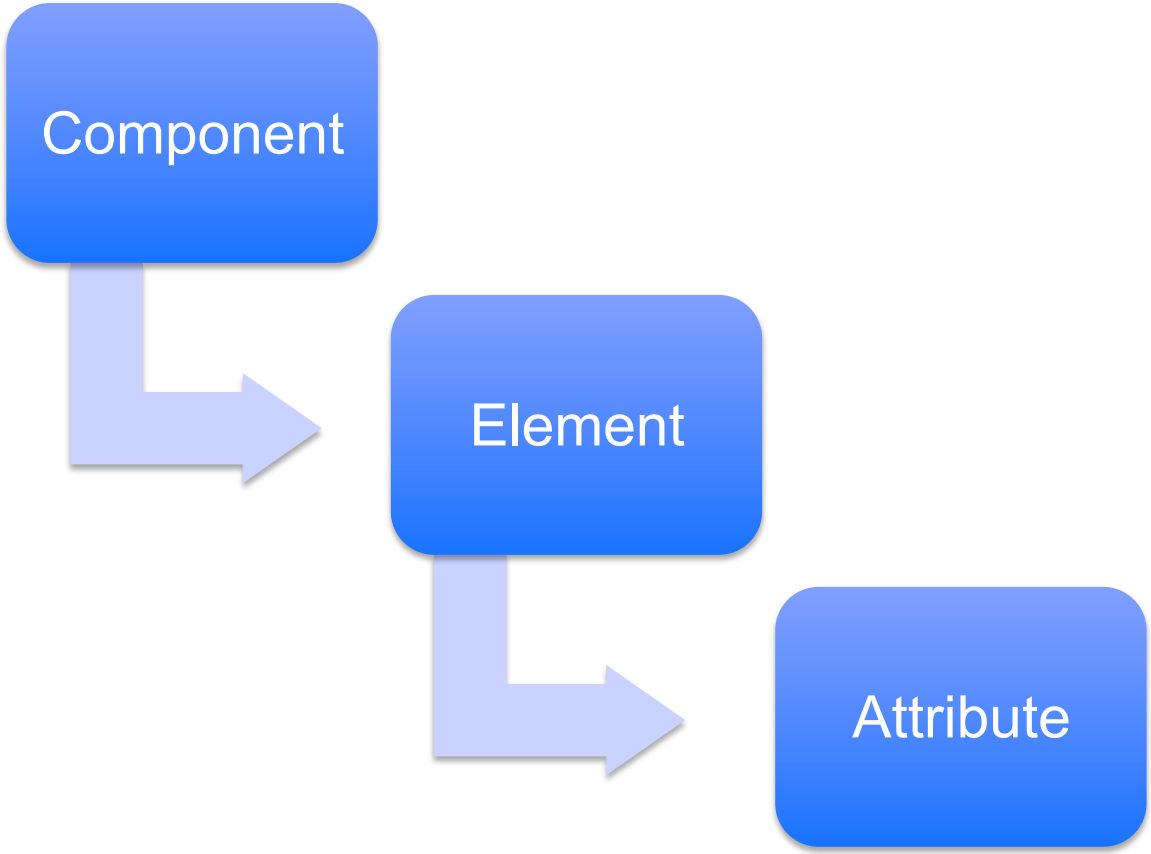
How do your components map?

Before you publish, you can check how your components map to VLO facets!

http://lux13.mpi.nl/clarin/vlo/mapping/index.html#check

Beyond the rules:
Modelling best practices

Description levels



Components

Represent a **resource** or an aspect thereof (e.g. *actor* or *technical metadata*)

- Can be '**inline**' or **referenced**. Functionally equivalent, but referenced means reusable.

- Consider assigning a **container data category**

Elements

Describe a **property** of a resource (aspect)

- Choose an appropriate **value scheme**
  - Default is *string* (text) but not always most applicable
  - Controlled vocabulary if applicable
- **Always** assign a **data category** if you want the element to be picked up by exploitation software
- Consider making multilingual

Attributes

"Annotate" components and elements, therefore **'meta-metadata'**

- Need for this is rare (especially on components)
- Fewer features than elements
- Often can be re-modelled as elements
- Useful for mimicking other formats

Component, element, attribute **naming**

- Use **prefixes and postfixes** judiciously
  - Names do not have to be unique
  - Preferably don't use to describe
    - Variation (e.g. *OralHistoryInterviewDANS*)
    - Revision (e.g. *TxtCrpsProfile-20110928*)
    - Delta (e.g. *access-addedfield*)
  - Use **description** or **group name** instead

Component, element, attribute **naming**

- **Spaces** are not allowed
  - Underscores? Hyphens? CamelCase?
  - Choose a **short name** and use *description* for a full name

More information on CMDI best practices

http://www.clarin.eu/cmdi

http://clarin.eu/faq-page

Thank you for your attention!
Questions?

Questions/Remarks can be sent to:
cmdi@clarin.eu