It's fun to play with the

Component **M**eta**D**ata **I**nfrastructure

Using component metadata

Dieter Van Uytvanck
Max Planck Institute for Psycholinguistics
Dieter.VanUytvanck@mpi.nl

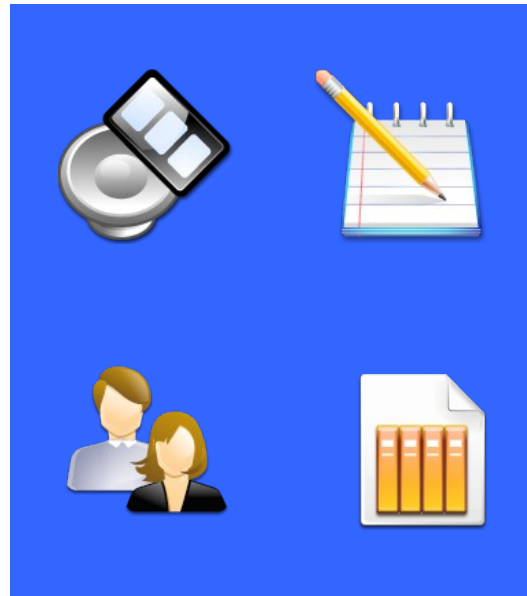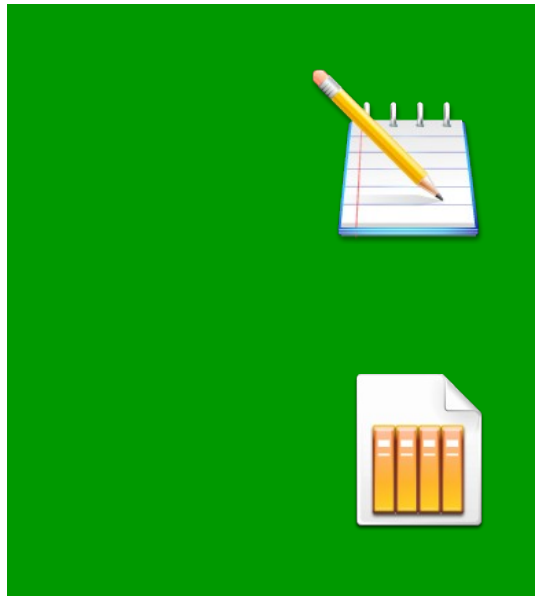**CLARIN-FR tutorial**
Paris
2011-21-01

Overview

- Traditional metadata
- Component metadata
- Data categories
- The big picture
- In practice:
    - Building components
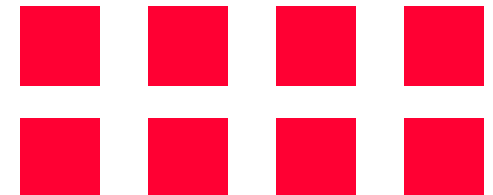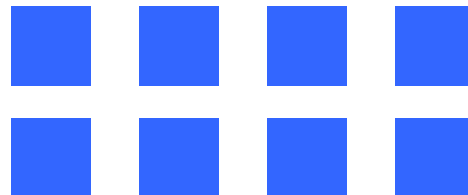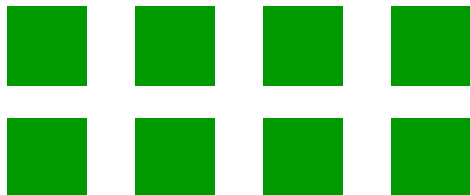   - Using components

Traditional Metadata

project 1

project 2

project 3

Traditional Metadata: problems

- Lack of flexibility
  - Too many fields...
  - ... but not the ones I am looking for!
- Lack of interoperability
  - My metadata does not work with your infrastructure!
 - Nederland? Netherlands? The Netherlands? Holland? NL?

Context

- Other Metadata Infrastructures in our domain:
  - IMDI, OLAC/DC, TEI header
- Problems:
  - Inflexible: too many (IMDI) or too few (OLAC) fields
  - Limited interoperability
  - Problematic (unfamiliar) terminology for some sub-communities.
 - etc.

CLARIN Project - CMDI

- Metadata infrastructure based on a "Component Metadata Model"
- Aims
  - Flexibility
    - Researcher should themselves decide what metadata fits their needs
    - Offer ready made metadata components
    - Allow creation of new metadata components needed
  - Interoperability built-in
  - Complete Infrastructure: software for editing, harvesting, exploitation
  - Compatibility with existing frameworks: OLAC, IMDI
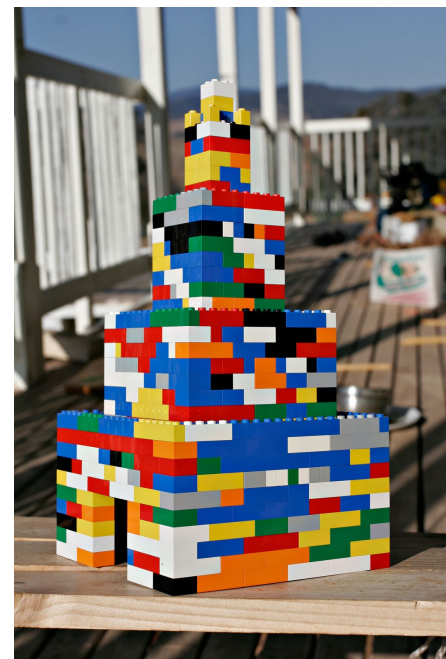
Component Metadata
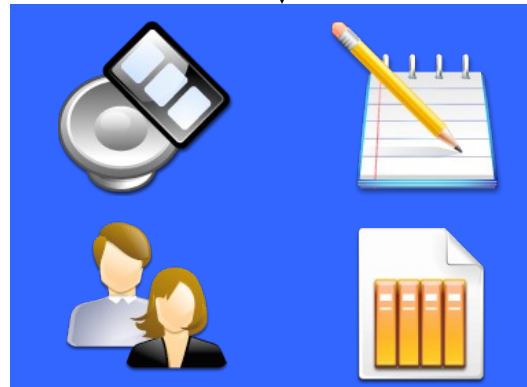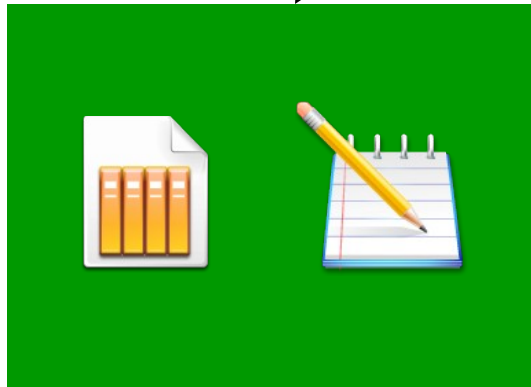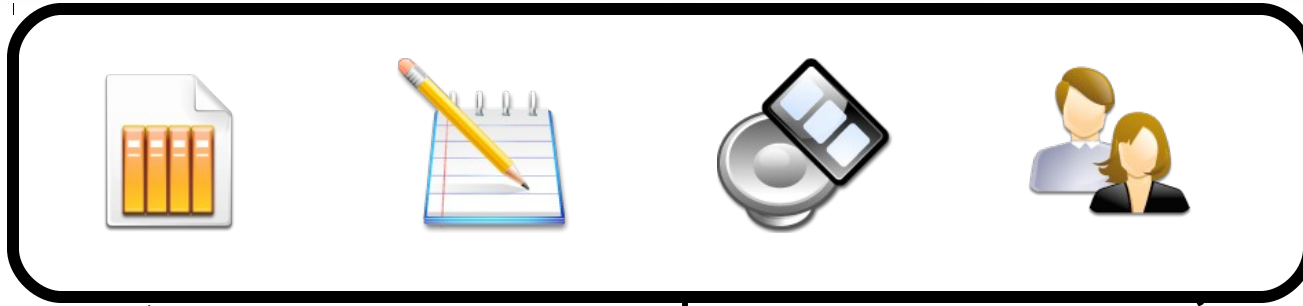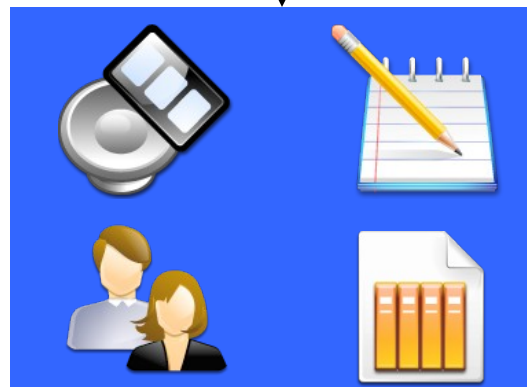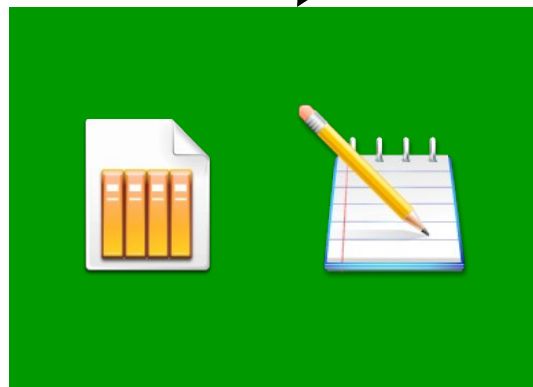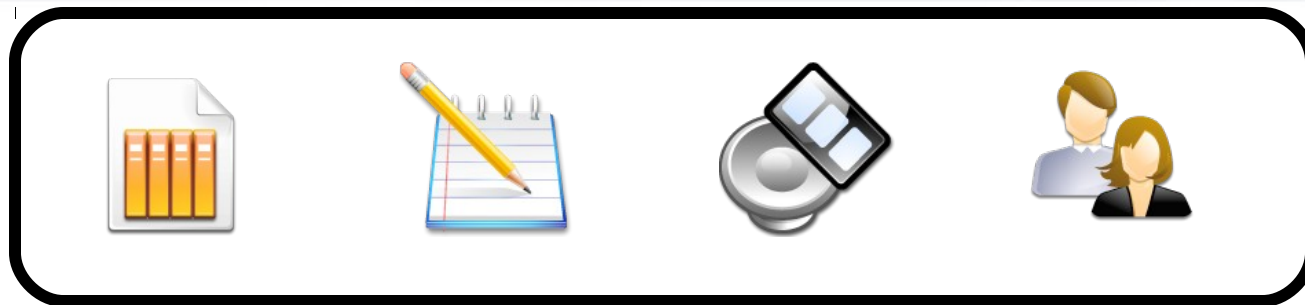

project 1

project 2

project 3

Some terminology

- Element = atomic unit (a "field") – e.g. recording date
- Component = set of elements – e.g. Actor
- Profile = set of components – e.g. OLAC profile
- Instance = one metadata description – e.g. myresource.cmdi

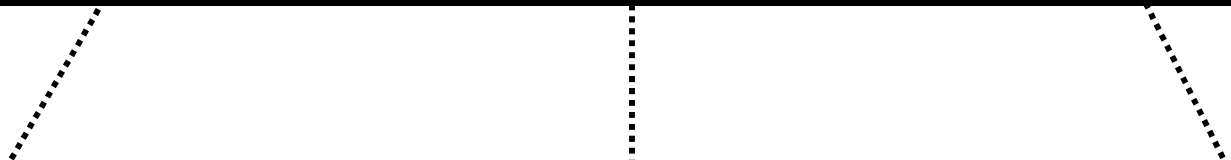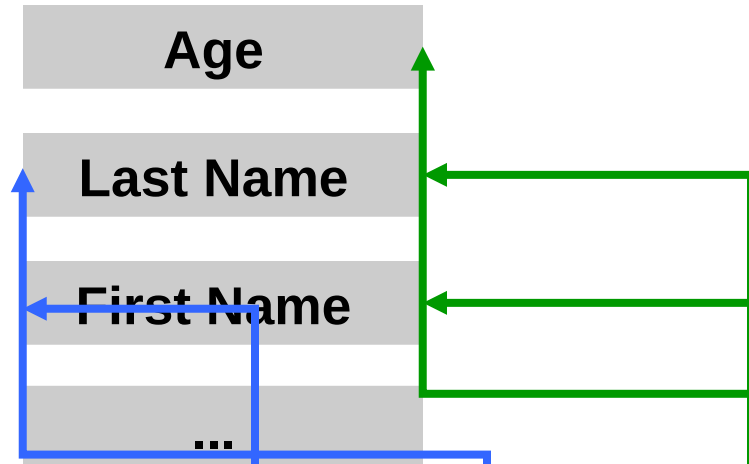- Schema = technical (formal) grammar describing a profile – e.g. olac.xsd

Data Categories

Data Categories

Age

Last Name
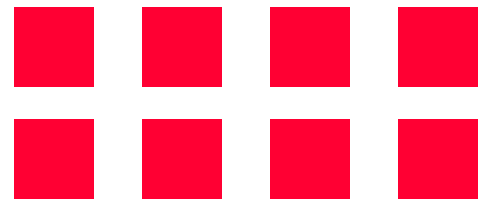
First Name

...

The big picture

Data Category
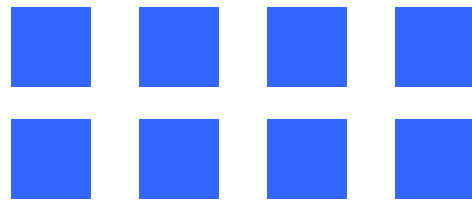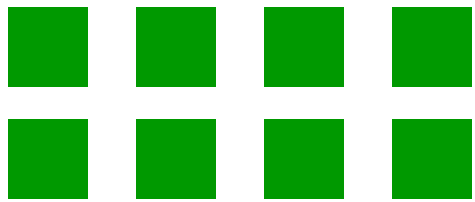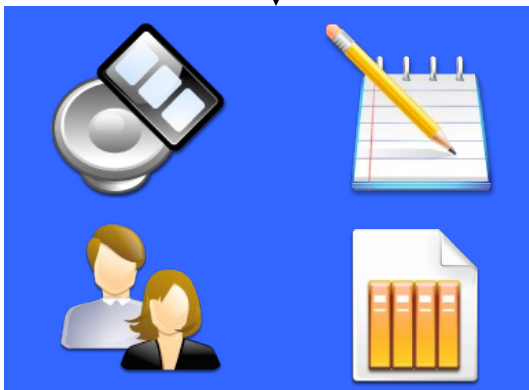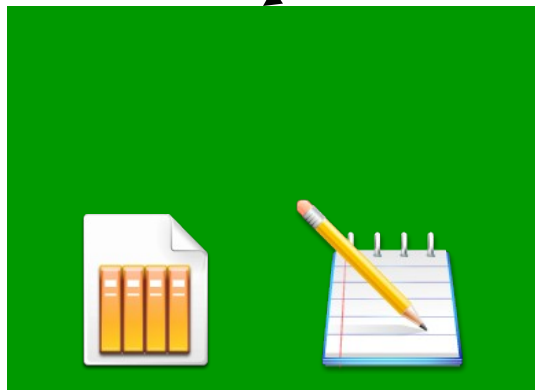
Building a component

Actor

firstName   lastName

ActorLanguage

languageCode   languageName

```
<CMD_Component name="Actor">
  <CMD_Element name="firstName"
ValueScheme="string"/>
  <CMD_Element name="lastName"
ValueScheme="string"/>
  <CMD_Component name="ActorLanguage">
   <CMD_Element name="LanguageCode"
ValueScheme="string" />
    <CMD_Element name="LanguageName"
      ValueScheme="string"
ConceptLink="http://www.isocat.org/datcat/DC-
1766"/>
  </CMD_Component>
</CMD_Component>
```

Using a component


```
[...]
<Actor>
  <firstName>Louis</firstName>
  <lastName>Couperus</lastName>
  <ActorLanguage>
        <LanguageCode>nld</LanguageCode>
        <LanguageName>Dutch</LanguageName>
  </ActorLanguage>
</Actor>
[...]
```

A close look at a CMDI file

- A toy example:
  - http://www.clarin.eu/cmd/example/example-md-instance.cmdi

- A corpus description:
  - http://www.clarin.eu/cmd/example/example-phonological-corpus.cmdi

Process overview

- Check the Component registry:
  - Any profile that fits your needs?
  - If not:
    - Any component that fits your needs?
    - If not:
      - Create your own component!
      - Looking for a data category that is not there?
        - Create a new data category!
    - Combine components together in a profile
- Start Arbil or your XML-editor
  - Select the profile/XSD that suits your needs
  - Create metadata instances

Conclusions

- Building your own components and profiles is already possible

- Creating CLARIN metadata descriptions too

- This is **not** the complete infrastructure:

  - Format will be supported in the future

  - Component Registry and Arbil will be extended

    - Please, send your feedback!

  - To be expected: user friendly

    - Browsers / Repositories

    - search engines

Where to get the software?

- http://www.clarin.eu/cmdi

Where to get support?

- cmdi@clarin.eu

There's no need to be unhappy!

Thank you for your attention