CuPED:
Software Demonstration

Christopher Cox
University of Alberta

Andrea L. Berez
U California, Santa Barbara

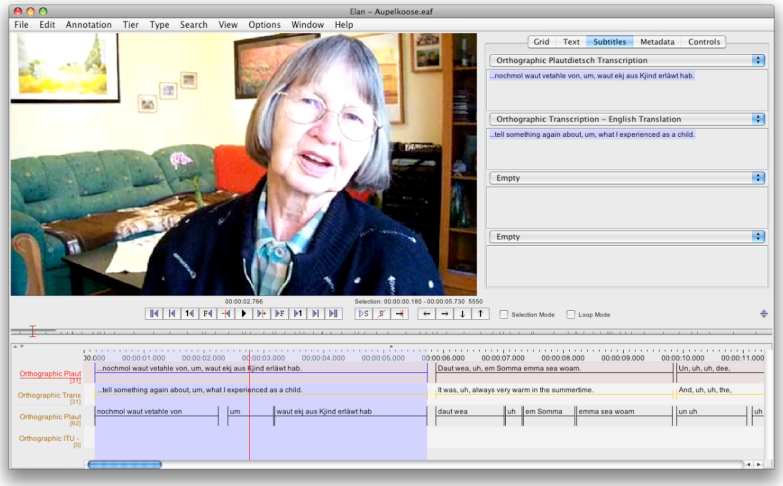
1st International Conference on Language Documentation and Conservation, University of Hawaii, 14 March 2009.
Icon designed by Stephanie Morse.

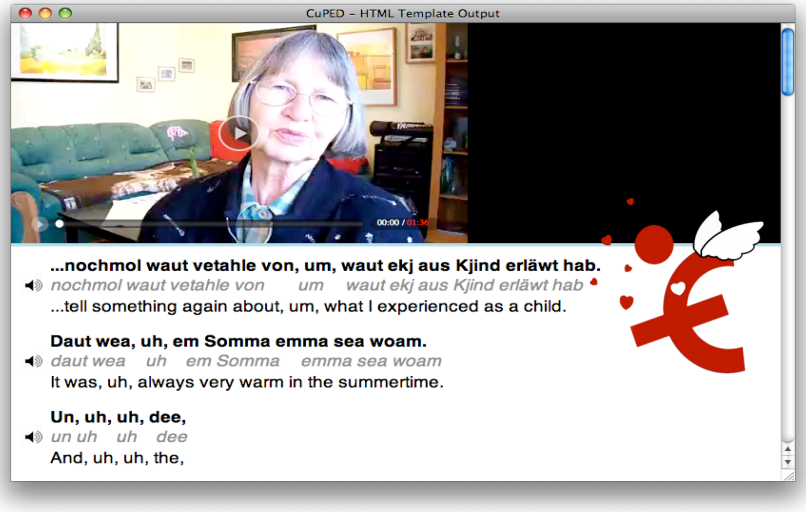
What is CuPED?

- **Cu**stomizable **P**resentation of **E**LAN **D**ocuments
- **Easily** transforms ELAN-annotated video or audio stream into an **attractive, user-customizable, interactive display**
- Deliverable by web, CD, print ...

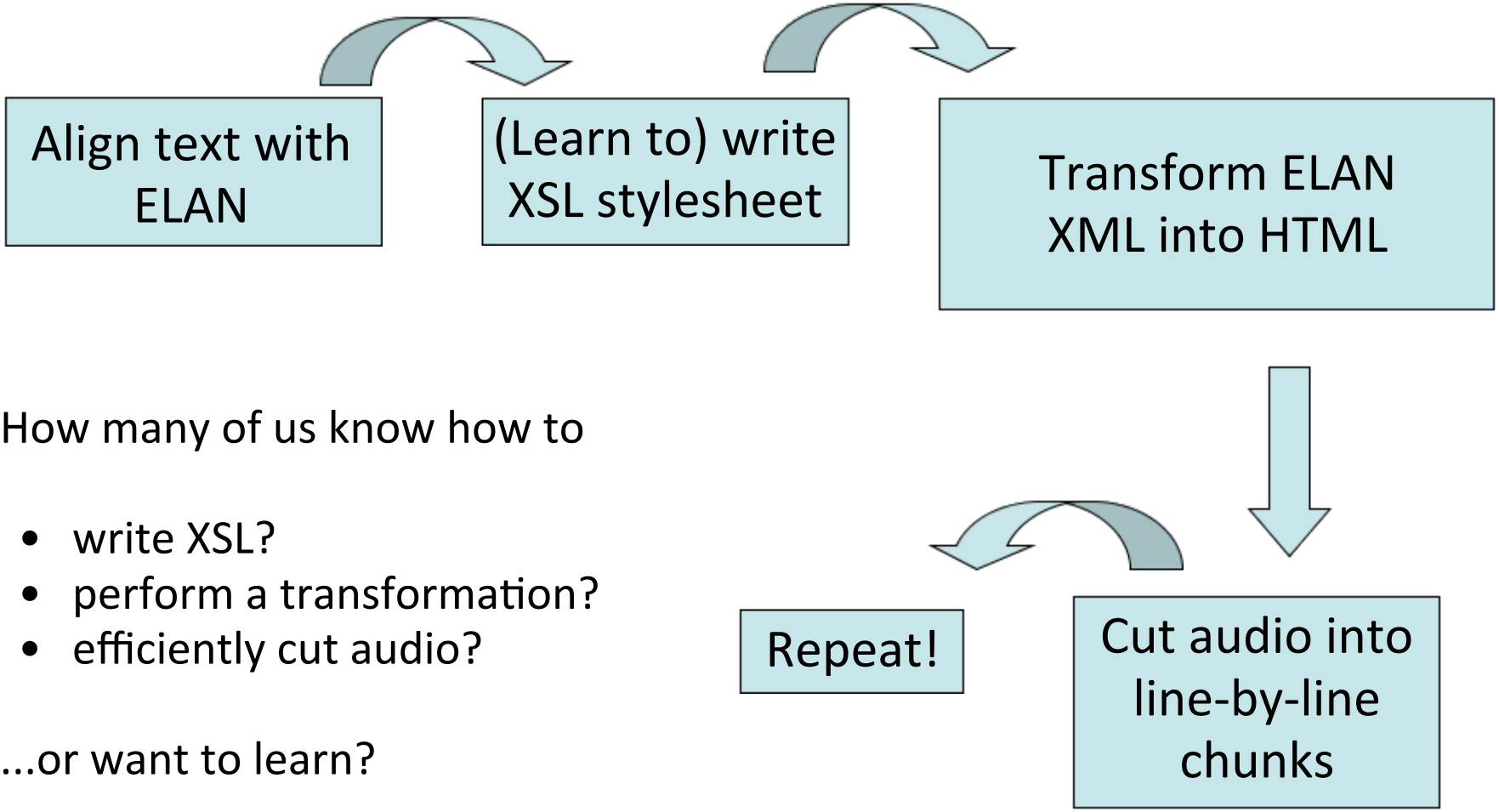
Why do we need **CuPED**?

- ELAN's **archival output** is becoming the standard for annotated media
- But ELAN's interface is functional, not eye-catching and fun
- Teaching/presentation materials integral to language work
- Motivates the creation of archival annotations by **facilitating easy presentation versions**





An example of a previous workflow



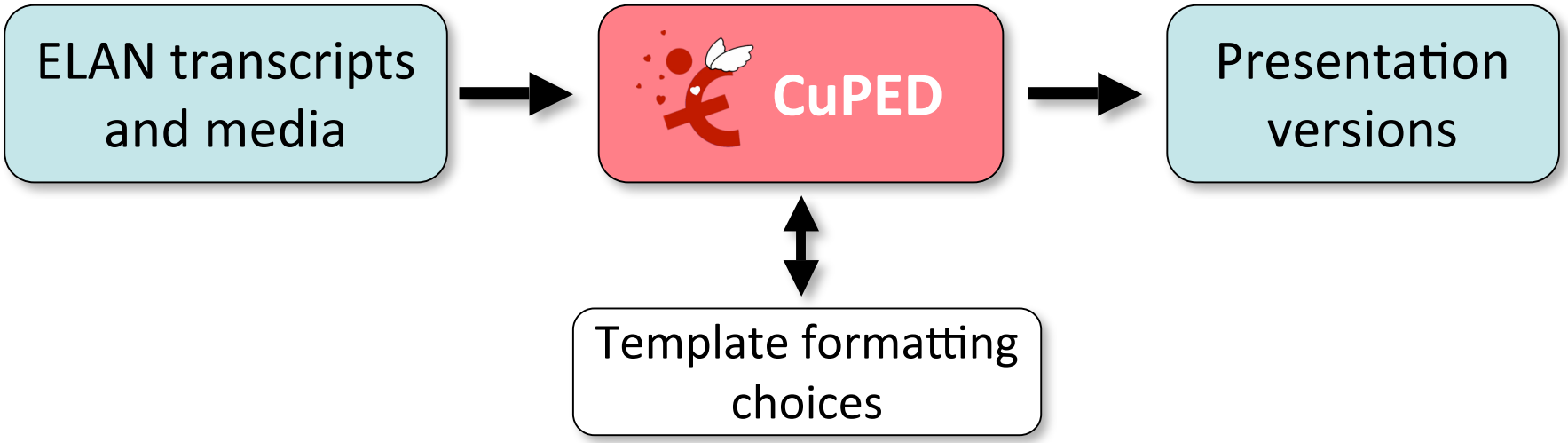
How many of us know how to

- write XSL?
- perform a transformation?
- efficiently cut audio?

...or want to learn?

How it works.

- CuPED adopts a **template-driven approach** to creating presentation versions of ELAN-annotated audio or video:

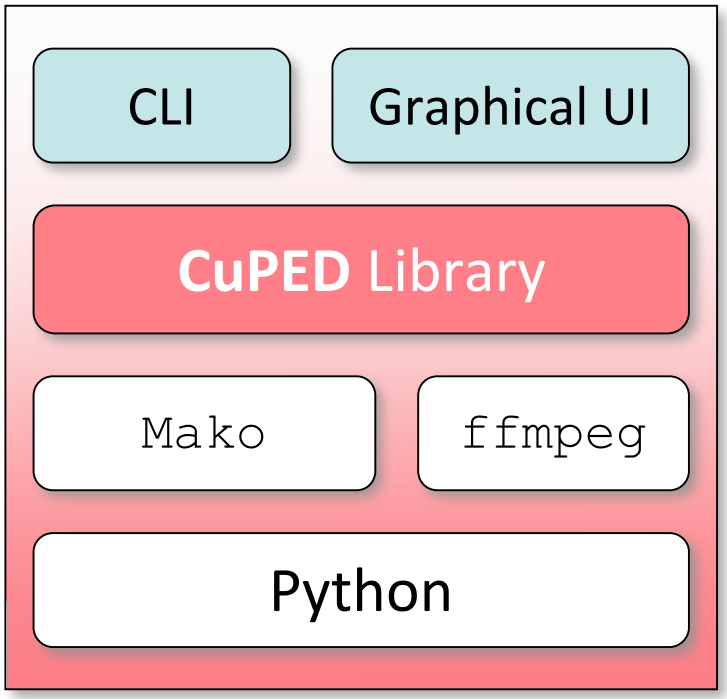


- Templates provide the **basic structure** into which the contents of transcripts are inserted, and present the user with **choices** as to how the contents should be formatted

Under the hood...

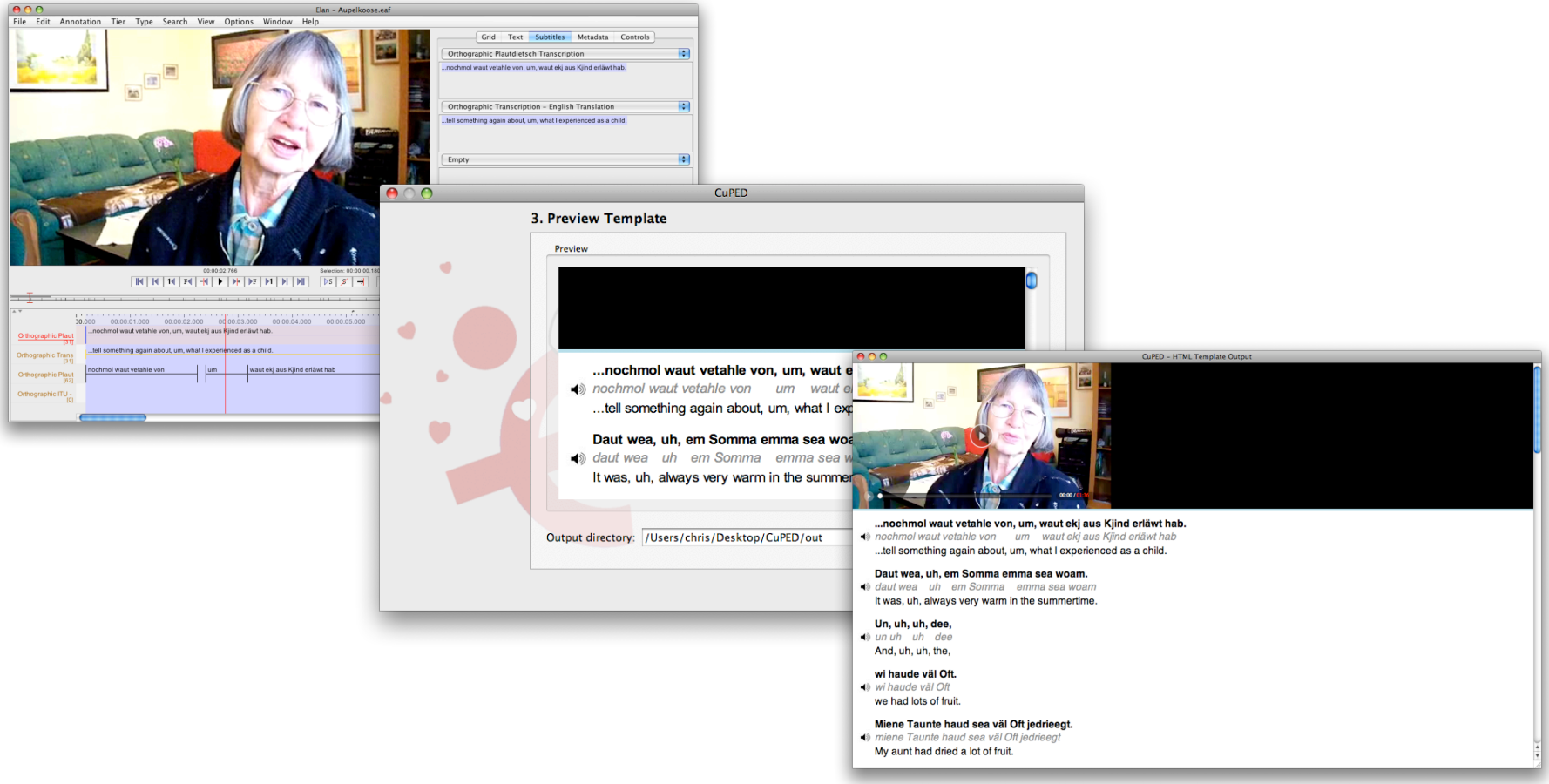
- CuPED thus **takes care of media conversion, clip extraction, tier formatting**, and other, presentation-related work **transparently**, allowing you to concentrate on annotation.

- Looking under the hood:
  1. **Written in Python**; currently runs on several platforms (precompiled versions for Windows and Mac OSX)
  2. **Uses** several actively-developed, **open-source components** to process media (`ffmpeg`) and templates (`Mako`)
  3. **Extensible**: CuPED comes with a set of example templates, but eager users can also write their own (XML)



Example: Using **CuPED**

- Take **CuPED** for a test drive:



Conclusions

- CuPED is under active development, with the existing code being improved upon and new features being added:
  - **Batch processing mode**: Processing many transcripts with one pre-configured template
  - **Command-line interface** to CuPED for simpler integration into server-side workflows
  - **New templates**: LaTeX output for professional-quality typesetting, turning ELAN transcripts into printed materials easily

- Following ELAN's example, CuPED is made **freely available** as **open source software** under the GNU General Public License.
  - Would hope that CuPED might help reduce burden of presenting ELAN transcripts and media, allowing more time to be spent on the development of primary materials themselves.

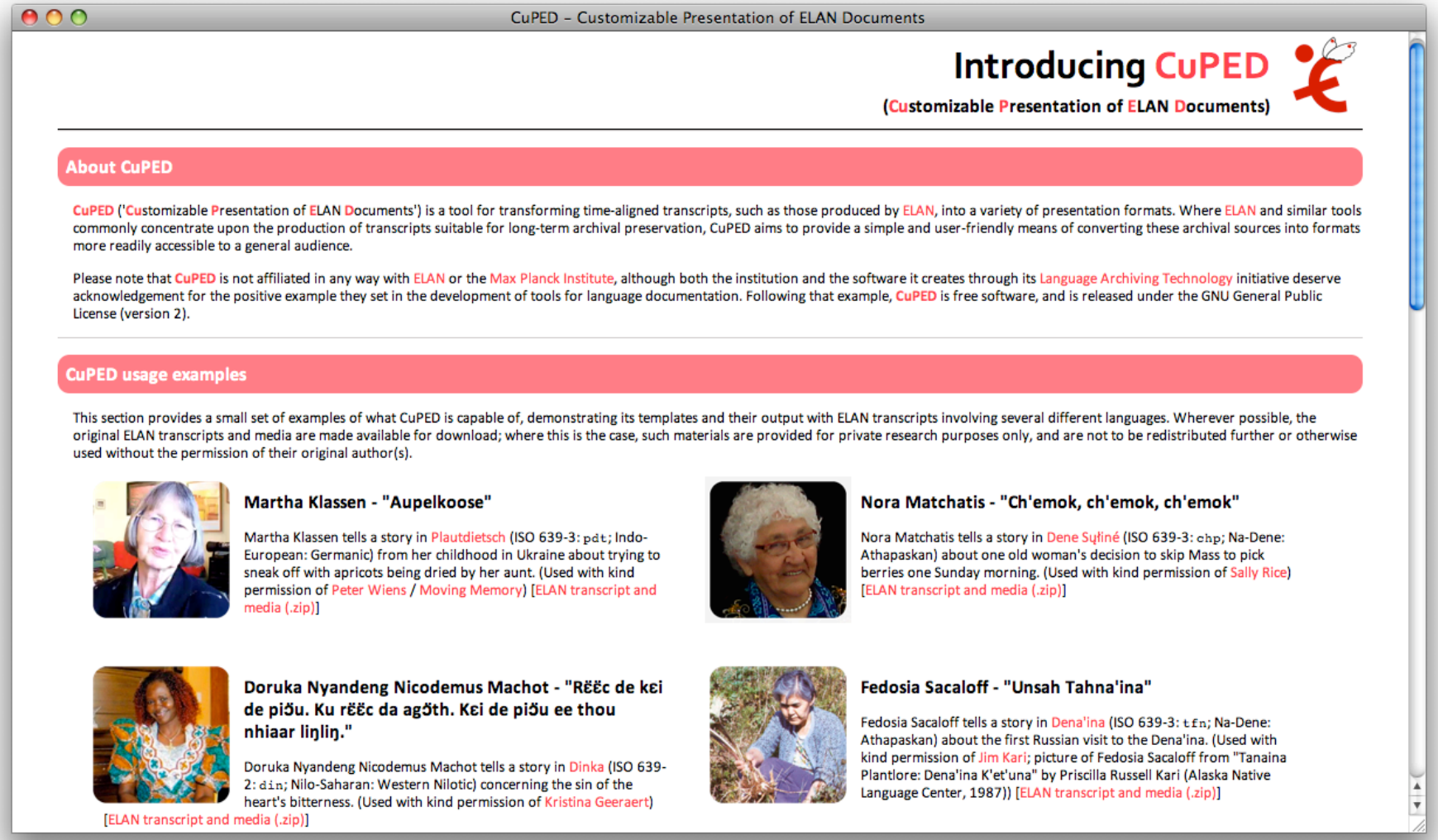
Thanks!

Acknowledgements

- **Stephanie Morse**:
  - CuPED artwork
- **Kristina Geeraert**, **Jim Kari**, **Sally Rice**, **Peter Wiens**:
  - Example transcripts / media used in CuPED development
- **Department of Linguistics**, **University of Alberta**:
  - Server space and bandwidth for CuPED development

- **U.S. National Science Foundation**:
  - Funding through the Dena'ina Archiving, Training and Access Project (Grant NSF-OPP 0326805)
- **Social Sciences and Humanities Research Council of Canada**, **Killam Trusts**, **Department of Canadian Heritage**:
  - Research funding

Social Sciences and Humanities Research Council of Canada
Conseil de recherches en sciences humaines du Canada
Canada
NSF
Killam Trusts



CuPED - Customizable Presentation of ELAN Documents

http://www.artsrn.ualberta.ca/cdcox/cuped/

CuPED vs. XSLT

- **Q:** Why not just use XSL(T)?
  1. XSLT doesn't take care of **media conversion** for you. One could still use XSLT to process templates, of course, but a second stage of interpretation beyond what XSLT itself provides would be needed to process any associated media.
  2. XSLT doesn't necessarily provide the **simplest access to ELAN data structures**. Finding all annotations that overlap in time with a given annotation, for instance, may be non-trivial with XSLT.
  3. As a consequence of providing direct access to the original ELAN XML, XSLT may prove **brittle** in the face of changing file formats. If the ELAN transcript file format changes, your XSL template may no longer work.
  4. XSLT **requires XML input**: if we wanted to extend CuPED to handle other, non-XML transcript formats (e.g. CLAN transcripts, Praat TextGrids), XSLT wouldn't be able to help.

ELAN integration

- **Q:** Why not integrate CuPED into ELAN itself?
  - At present, ELAN provides **no media conversion facilities**: it displays and annotates media, but does not edit them. Given the cross-platform nature of ELAN, and the platform-specificity of many media editing components, adding media conversion capabilities to ELAN may result in fewer platforms being supported, and further complicate building new versions of ELAN.
  - Nevertheless, basic integration may be possible: since CuPED is written in Python, one might **port the source code to Jython**, a Java-based implementation of Python, and thus integrate it with ELAN.
    - (Certain features may suffer in the process: CuPED is able to provide more reliable previews of web content than would likely be possible in "pure" Java)

CuPED templates

- **Q:** What exactly *are* CuPED templates?
  - CuPED templates consist of two things: a set of **input files** (e.g. images, data, etc.) processed by Mako, and an **XML file** which says what kinds of information the user should provide (e.g. transcripts).

```
%for tier in transcript.tiers:
    * ${tier.name}
%endfor
```

* Tier one name
* Tier two name

...

* Tier *n* name

CuPED templates

- Producing basic HTML:

```
<ul>
%for tier in transcript.tiers:
    <li>${tier.name}</li>
%endfor
</ul>
```

```
<ul>
  <li>Tier one name</li>
  <li>Tier two name</li>
  ...
  <li>Tier n name</li>
</ul>
```

CuPED templates

- Producing LaTeX lists:

```
\begin{itemize}
%for tier in transcript.tiers:
    \item{${tier.name}}
%endfor
\end{itemize}
```

```
\begin{itemize}
  \item{Tier one name}
  \item{Tier two name}
  ...
  \item{Tier n name}
\end{itemize}
```