Maik Stührenberg

PROCESSING MULTIMODAL DOCUMENTS

Aarhus, Denmark
HOW I FEEL DURING THIS CONFERENCE UP TO NOW...
SOME WORDS OF CAUTION

- The following talk will be about technical issues
- If you are here by accident – leave the room now!

...otherwise enjoy the show!
STRUCTURE OF THIS TALK

1 Introduction
2 Multimodal documents
3 Standoff annotation
4 XStandoff
   ■ Segmentation
   ■ Annotation
5 Conclusion
1 Introduction

2 Multimodal documents

3 Standoff annotation

4 XStandoff

5 Conclusion
The title of this talk is “Processing multimodal documents”...

- What do I mean with “processing”?
- What are multimodal documents?
THE CONCEPT OF ANNOTATION

- In Computational Linguistics (and especially in Text-technology), processing heavily depends on annotation.
- Annotation is the concept of adding relevant information to the primary data (the information to be annotated).

Example

Hey Paul! Would you give me the hammer?

POS annotation (Stanford NLP tagger, txt output)

Hey_NNP Paul_NNP !_.
Would_MD you_PRP give_VB me_PRP the_DT hammer_NN ?_.

INSTITUT FÜR DEUTSCHE SPRACHE
For over 15 years, XML is the metadata standard for annotating information of various kinds

POS annotation (Stanford NLP tagger, XML output)

```xml
<pos>
  <sentence id="0">
    <word wid="0" pos="NNP">Hey</word>
    <word wid="1" pos="NNP">Paul</word>
    <word wid="2" pos=">!</word>
  </sentence>
  <sentence id="1">
    <word wid="0" pos="MD">Would</word>
    <word wid="1" pos="PRP">you</word>
    <word wid="2" pos="VB">give</word>
    <word wid="3" pos="PRP">me</word>
    <word wid="4" pos="DT">the</word>
    <word wid="5" pos="NN">hammer</word>
    <word wid="6" pos=".”>?</word>
  </sentence>
</pos>
```
The data structure of an XML instances resembles a tree

```
Hey, Paul! Would you give me the hammer?```

```
pos

sentence

word  word  word  word  word  word  word  word  word

Hey  Paul  !  Would  you  give  me  the  hammer  ?```
**BENEFITS OF XML ANNOTATION**

- XML as a meta language allows to create markup languages for any given purpose
- XML is an open standard
- XML instances are text files
- XML fully supports Unicode
- Accompanying standards such as XSLT (Transformation), XPath (Traversal), and XQuery (Query) and a large number of (freely available and/or Open Source) tools are available as well

  Hint: You often use XML without knowing it (DOCX, PPTX, XLSX, ...
1 Introduction

2 Multimodal documents

3 Standoff annotation

4 XStandoff

5 Conclusion
We call digital documents, that combine multiple information encodings, that is (selectable) text and visuals or other non-textual encoding *multimodal documents*

- Multiple representations are not just alternative encodings of the very same information
- Different representations may be related to each other
- These relations are interesting research objects

Let’s start with a simple example...
1. Use the mode dial (1) to select the 'A' or 'P' mode
2. Press the shutter (2) half-way down, to focus and to release the built-in flash (4) Alternatively, use an external flash connected to the hotshoe (3)
3. Press the shutter (2) full-way down to take the picture
Typical instructions for taking pictures with a digital camera contain both images and proper written instructions.

Text parts refer to regions of the picture (in this example by numbers).

Regions of the picture depict the corresponding controls of the camera to interact with in real world.

An integrative serialization format would allow to link the regions of the picture directly to the corresponding text string.

Let’s move on to a more interesting example...
Another situation in which team A is not able to finish its move: Y tries to pass the ball through the small gap to X (instead of passing it to G) while Z is unintentionally obstructing Y’s way. But before the ball reaches X, D intercepts and passes the ball to A.

See http://spielverlagerung.de for real world examples
Another situation in which team A is not able to finish its move: Y tries to pass the ball through the small gap to X (instead of passing it to G) while Z is unintentionally obstructing Y’s way. But before the ball reaches X, D intercepts and passes the ball to A.

There are different information items here:
- ...about teams
- ...about players
- ...about places
Another situation in which team A is not able to finish its move: Y tries to pass the ball through the small gap to X (instead of passing it to G) while Z is unintentionally obstructing Y’s way. But before the ball reaches X, D intercepts and passes the ball to A.

Not every information is represented in both information encodings:

- Where is the player named “G”?
- Where is the place called “small gap”?
If we want to annotate a multimodal document, we have to split up text and images.

A simple inline annotation of the text could use the following elements and attributes:

```xml
<text>
Another situation in which <team name="teamA">team A</team> is not able to finish its move: <player name="Y">Y</player> tries to pass the ball through <place name="gap">the small gap</place> to <player name="X">X</player> (instead of passing it to <player name="G">G</player>) while <player name="Z">Z</player> is unintentionally obstructing <player name="Y">Y</player>’s way. But before the ball reaches <player name="X">X</player>, <player name="D">D</player> intercepts and passes the ball to <player name="A">A</player>.
</text>
```
To annotate the relevant parts of the image we make use of the following assumptions:

- Parts of the full graphic can be selected by using a coordinate system.
- Basic shapes can be used to ease the selection.
- For example, a circle can be described by a coordinate pair $x_{center}$, $y_{center}$ and the radius $r$.
- The dot depicting player Y can be described by $x_{center}=138$, $y_{center}=278$, and $r=9$ (in px, starting from top left corner).
HTML’s image map could be used to serialize visuals:

1. the `img` element provides a reference to the file containing the image
2. the `map` element contains the definition of an image map
3. each `area` element defines a part of the image as sensitive and provides a link target

```html
<img src="soccer.png" width="362" height="321" border="0" alt="map" usemap="#map"/>
<map name="map">
  <area shape="circle" coords="138,278,9" href="#player_Y" alt="Player Y" title="Player Y"/>
</map>
```
INTERMEDIATE RESULTS

Where are we now?

- We can annotate text and images separately
- Text is annotated inline, images have to be annotated standoff

Where do we want to go from here?

We would like to have an integrative serialization format supporting
- multimodal documents and
- multiple annotation layers
1 Introduction

2 Multimodal documents

3 Standoff annotation

4 XStandoff

5 Conclusion
What is Standoff Annotation?

**Definition**
Separation of primary data and the markup, and the usage of pointing mechanisms to link between the two.

**Building blocks of a standoff serialization format**
1. Segmentation of the primary data
2. Linking between segments and annotation
## PROS AND CONS

### Advantages
- Multiple annotation layers supported
- Scalable
- Different serializations possible

### Disadvantages
- Not very human-readable
- Robustness regarding primary data integrity depends on the approach chosen
1 Introduction

2 Multimodal documents

3 Standoff annotation

4 XStandoff
   - Segmentation
   - Annotation

5 Conclusion
Originally developed as Sekimo Generic Format (SGF) in the Sekimo project (Stührenberg and Goecke 2008)

- Standoff *meta* annotation format
- Supports multiple primary data files
- Segmentation mechanism for various primary data types
- Level (concept)/Layer (serialization) distinction
- Multiple annotation levels – no restriction about markup inventory
- ISOcat attributes for imported markup layers
- XSD 1.1 schema (since XStandoff 2.1)
- Accompanied by the XStandoff Toolkit (Stührenberg and Jettka 2009)
XStandoff structure (w/o metadata)

```xml
<corpusData xml:id="c1">
  <primaryData xml:id="p1">
    <!-- reference to primary data - multiple occurrences -->
  </primaryData>
  <segmentation>
    <!-- segments -->
  </segmentation>
  <annotation>
    <level>
      <layer>
        <!-- annotation layer(s) - multiple occurrences-->
      </layer>
    </level>
    <!-- additional annotation level -->
  </annotation>
  <!-- additional corpusData entries -->
</corpusData>
```
XSTANDOFF 101

XStandoff structure (w/o metadata)

<corpusData xml:id="c1">
  <primaryData xml:id="p1">
    <!-- reference to primary data - multiple occurrences -->
  </primaryData>
  <segmentation>
    <!-- segments -->
  </segmentation>
  <annotation>
    <level>
      <layer>
        <!-- annotation layer(s) - multiple occurrences-->
      </layer>
    </level>
    <!-- additional annotation level -->
  </annotation>
  <!-- additional corpusData entries -->
</corpusData>
XSTANDOFF 101

XStandoff structure (w/o metadata)

<corpusData xml:id="c1">
  <primaryData xml:id="p1">
    <!-- reference to primary data - multiple occurrences -->
  </primaryData>
  <segmentation>
    <!-- segments -->
  </segmentation>
  <annotation>
    <level>
      <layer>
        <!-- annotation layer(s) - multiple occurrences-->
      </layer>
    </level>
    <!-- additional annotation level -->
  </annotation>
  <!-- additional corpusData entries -->
</corpusData>
XSTANDOFF 101

XStandoff structure (w/o metadata)

<corpusData xml:id="c1">
  <primaryData xml:id="p1">
    <!-- reference to primary data - multiple occurrences -->
  </primaryData>
  <segmentation>
    <!-- segments -->
  </segmentation>
  <annotation>
    <level>
      <layer>
        <!-- annotation layer(s) - multiple occurrences-->
      </layer>
    </level>
    <!-- additional annotation level -->
  </annotation>
  <!-- additional corpusData entries -->
</corpusData>
XSTANDOFF 101

XStandoff structure (w/o metadata)

```xml
<corpusData xml:id="c1">
  <primaryData xml:id="p1">
    <!-- reference to primary data - multiple occurrences -->
  </primaryData>
  <segmentation>
    <!-- segments -->
  </segmentation>
  <annotation>
    <level>
      <layer>
        <!-- annotation layer(s) - multiple occurrences-->
      </layer>
    </level>
    <!-- additional annotation layer -->
    <!-- additional corpusData entries -->
  </annotation>
</corpusData>
```
Segmentation in XStandoff depends on the primary data type – supported types in XStandoff 2.1 are:

- textual primary data
- multimedia primary data
- spatial primary data
- pre-annotated primary data (e.g. web pages)
BUILDING BLOCK 1: SEGMENTATION

Textual primary data

Textual primary data can be delimited by character positions:

This is a word
00|01|02|03|04|05|06|07|08|09|10|11|12|13|14

Serialization in XStandoff

<primaryData xml:id="txt">
  <xsf:primaryDataRef uri="soccer.txt" encoding="utf-8" mimeType="text/plain" start="0" end="265"/>
</primaryData>

<!-- [... ] -->

<segment xml:id="seg_text1" primaryData="txt" type="char" start="27" end="33"/>
Multimedia primary data (audio/video) can be delimited by time points.

Serialization in XStandoff:

```xml
<primaryData start="0" end="335587" unit="milliseconds" xml:id="pd1_video">
  <primaryDataRef uri="b1-video.mpg" mimeType="video/mpeg"/>
</primaryData>
<segment xml:id="seg2" primaryData="pd1_video" start="309924" end="310079"/>
```
BUILDING BLOCK 1: SEGMENTATION

Image primary data

Delimit spatial primary data with a coordinate system and basic shapes

Serialization of a circle in XStandoff

```xml
<xsf:primaryData xml:id="img" unit="pixels">
  <xsf:primaryDataRef uri="img.png" mimeType="image/png" width="824" height="679"/>
</xsf:primaryData>

<xsf:segment xml:id="seg1" type="spatial" primaryData="img" shape="circle" coords="312,651,23" name="X"/>
```

Serialization of a polygon in XStandoff

```xml
<xsf:segment xml:id="seg2" type="spatial" primaryData="img" shape="poly" coords="2400,125 2600,125 2400,945 2600,945" name="Y"/>
```

More complex shapes are serialized by Bezier curves
BUILDING BLOCK 1: SEGMENTATION

Describing parts of an image over time

By combining temporal and spatial segmentation, it is possible to annotate single actors, body parts, gestures, etc.

Serialization in XStandoff

```xml
<segment xml:id="s1" type="spatial" shape="poly" coords="0,11,20 3,4,30 1,2,30" start="00:20:00" end="00:20:01"/>
```

Alternatively, use segments instantiated by referring other segments

```xml
<segment xml:id="s1" type="spatial" shape="poly" coords="0,10,30 100,150,30 0,200,30 0,100,30"/>
<segment xml:id="s2" type="spatial" shape="poly" coords="10,10,40 110,150,40 10,200,40 110,100,40"/>
<segment xml:id="s3" type="seg" segments="s1 s2" name="AnkleLeft" mode="continuous" start="00:00:00" end="00:01:15"/>
```
While other standoff serialization formats use generic notation formats, XStandoff tries to stick as close as possible to an inline annotation.

**Inline**

```xml
<text>
Another situation in which <team name="teamA">team A</team> is not able to finish its move:
[player name="Y" team="teamA">Y</player> tries to pass the ball through <place name="gap">the small gap</place>
</text>
```

**XStandoff**

```xml
<text xsf:segment="seg1">
<team name="teamA" xsf:segment="seg2 seg19"/>
[player name="Y" team="teamA" xsf:segment="seg3 seg12"/>
<place name="gap" xsf:segment="seg4 seg18"/>
</text>
```
BRINGING IT ALL TOGETHER...

Linking annotations and segments

```xml
<corpusData>
  <primaryData xml:id="txt">
    <primaryDataRef uri="soccer.txt" encoding="utf-8" mimeType="text/plain" start="0" end="265"/>
  </primaryData>
  <primaryData xml:id="img" unit="pixels">
    <primaryDataRef uri="soccer.png" mimeType="image/png" width="362" height="321"/>
  </primaryData>
  <segmentation>
    <segment xml:id="seg3" start="66" end="67" primaryData="txt"/>
    <segment xml:id="seg12" type="spatial" shape="circle" coords="138,278,9" primaryData="img" name="Y"/>
    <segment xml:id="seg18" type="spatial" shape="poly" coords="142,246 167,185 188,194 163,255" primaryData="img" name="gap"/>
    <segment xml:id="seg19" type="seg" segments="seg12 seg13 seg14 seg17" mode="disjoint" name="Team A"/>
  </segmentation>
  <annotation>
    <level xml:id="soccer_a_b-level1">
      <layer>
        <text xsf:segment="seg1">
          <team name="teamA" xsf:segment="seg2 seg19"/>
          <player name="Y" team="teamA" xsf:segment="seg3 seg12"/>
        </text>
      </layer>
    </level>
  </annotation>
</corpusData>
```
Another situation in which **team A** is not able to finish its move: **Y** tries to pass the ball through the small gap to ...
EXPRESSING RELATIONS

Now that we have everything in one place, we can start annotating relations between different encodings.

The easy way: using both segment references and adding metadata

```xml
<player name="G" team="teamA" xsf:segment="seg6 seg17">
  <xsf:meta xmlns="http://www.tei-c.org/ns/1.0">
    <certainty locus="name" target="playerG" degree="0.9">
      <desc>Although the part of the graphic depicted with seg17 comes without a name tag, it is most likely that it depicts the player called ‘G’ in the running text (seg6).</desc>
    </certainty>
  </xsf:meta>
</player>
```
EXPRESSING RELATIONS

The more elaborate way: using an additional annotation layer

```xml
<!-- [...] -->
<level xml:id="soccer_relations">
  <layer>
    <relations>
      <isFormatOf xsf:segment="seg6 seg17" certainty="0.9"/>
    </relations>
  </layer>
</level>
```

- No restrictions about the relations and its serialization
- A possible format would be a serialization of the text-image relations described by Martinec and Salway (2005) or van Leeuwen (2005)
It is cumbersome to create XStandoff instances by hand
Therefore, we have developed the XStandoff Toolkit – a set of XSLT 2.0/XQuery 1.0 stylesheets, supporting...
- converting an inline annotation into an XStandoff instance containing a single standoff annotation layer
- merging two XStandoff instances over the same primary data file into a single instance
- deleting/removing annotation layers of an XStandoff instance
- converting an XStandoff instance into inline notation (including handling of possible overlaps)
- analyzing an XStandoff instance with multiple annotation layers
- visualizing an XStandoff instance (Jettka and Stührenberg 2011)

However, up to now, the XStandoff Toolkit does not support spatial segments
CURRENT STATE AND FUTURE WORK

- XStandoff 2.1 as format is stable and can be obtained at [http://xstandoff.net](http://xstandoff.net)
- What is needed to fully support annotation of multimodal documents, is a web-based annotation tool, capable of selecting both texts and parts of images via mouse-click
- There is already the web-based annotation tool Serengeti for texts using a former version of XStandoff
- A number of mouse-driven image selection tools are available that can be adopted
1 Introduction

2 Multimodal documents

3 Standoff annotation

4 XStandoff

5 Conclusion
CONCLUSION AND OUTLOOK

- If you create a corpus of (multimodal) documents, please use an open serialization format.
- Standoff notation as annotation method opens up a serialization for multimodal documents.
- XStandoff may be a suitable format for such a task since it already supports different segmentation methods and multiple annotation layers.
- Future work will have to focus on a user-friendly mouse-driven annotation tool for multimodal documents.
THANK YOU FOR YOUR ATTENTION!

stuehrenberg@ids-mannheim.de | maik@xstandoff.net