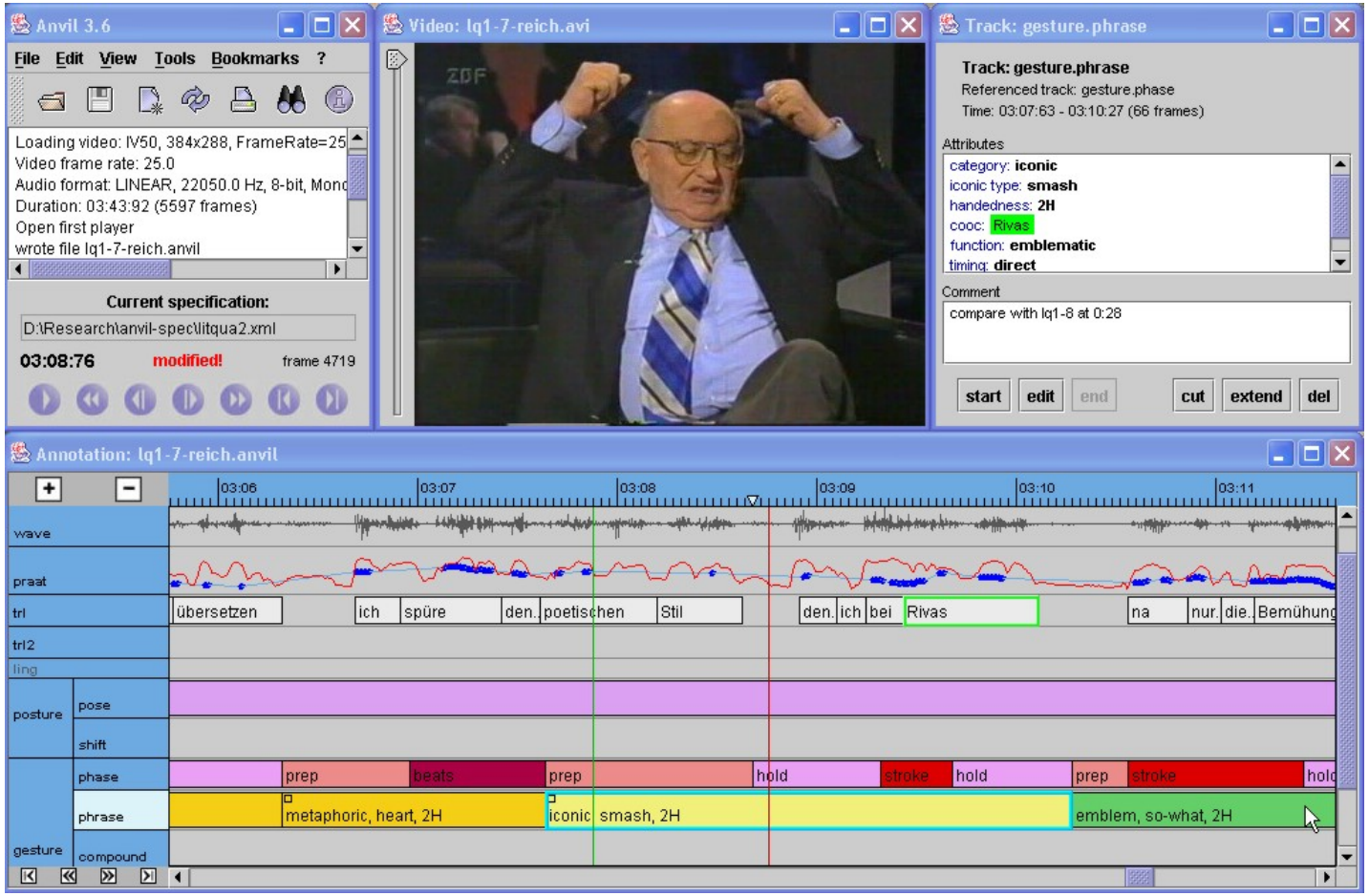
An exchange format for multimodal annotations

Thomas Schmidt, Susan Duncan, Oliver Ehmer, Jeffrey Hoyt, Michael Kipp, Dan Loehr, Magnus Magnusson, Travis Rose, Han Sloetjes

Background

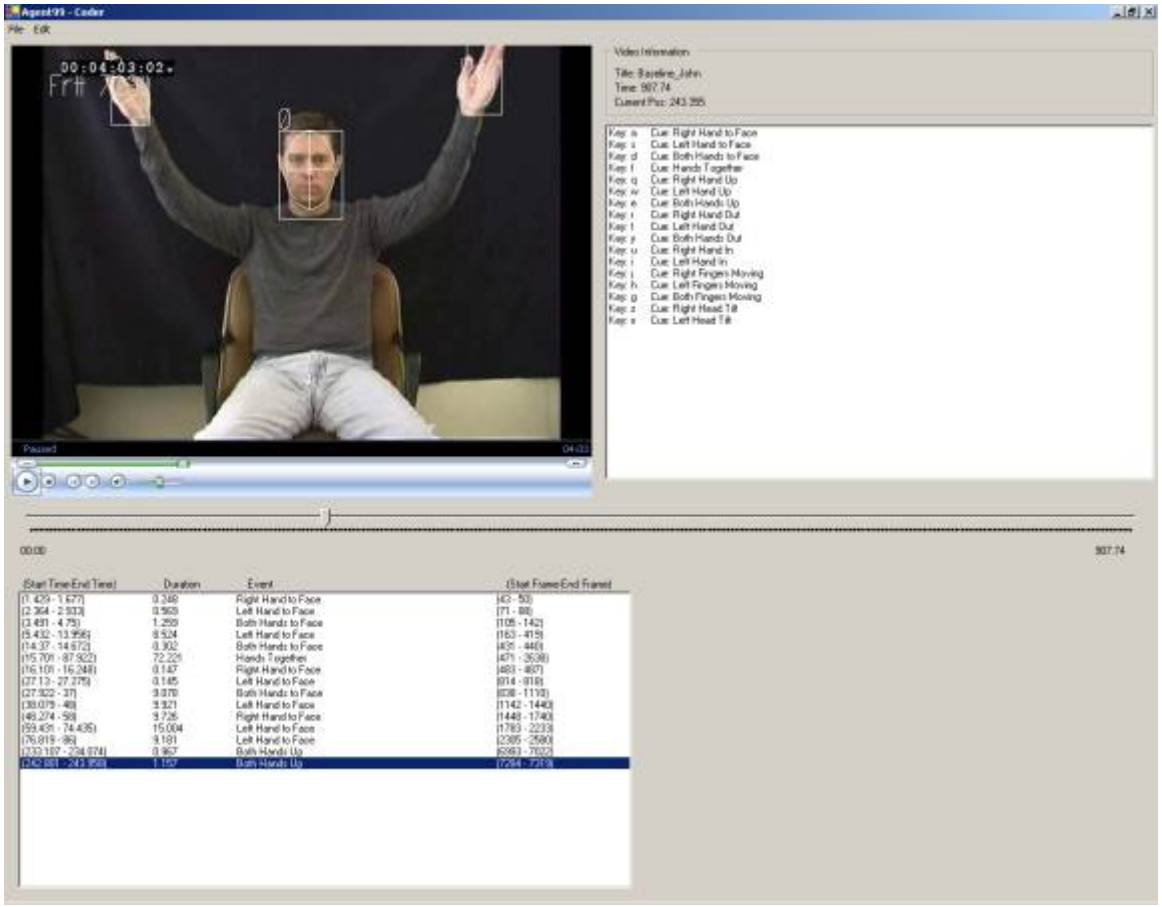
- International Society for Gesture Studies (ISGS)
  - 2005 Conference in Lyon (‚Interacting Bodies‘)
    - User workshop on ‚Multimodal Annotation Tools‘
  - 2007 Conference in Chicago (‚Integrating Gestures‘)
    - Developer workshop on ‚Annotation Interchange among Multimodal Annotation Tools’
    - Goal: Interoperability between existing tools

Tools (1): Anvil



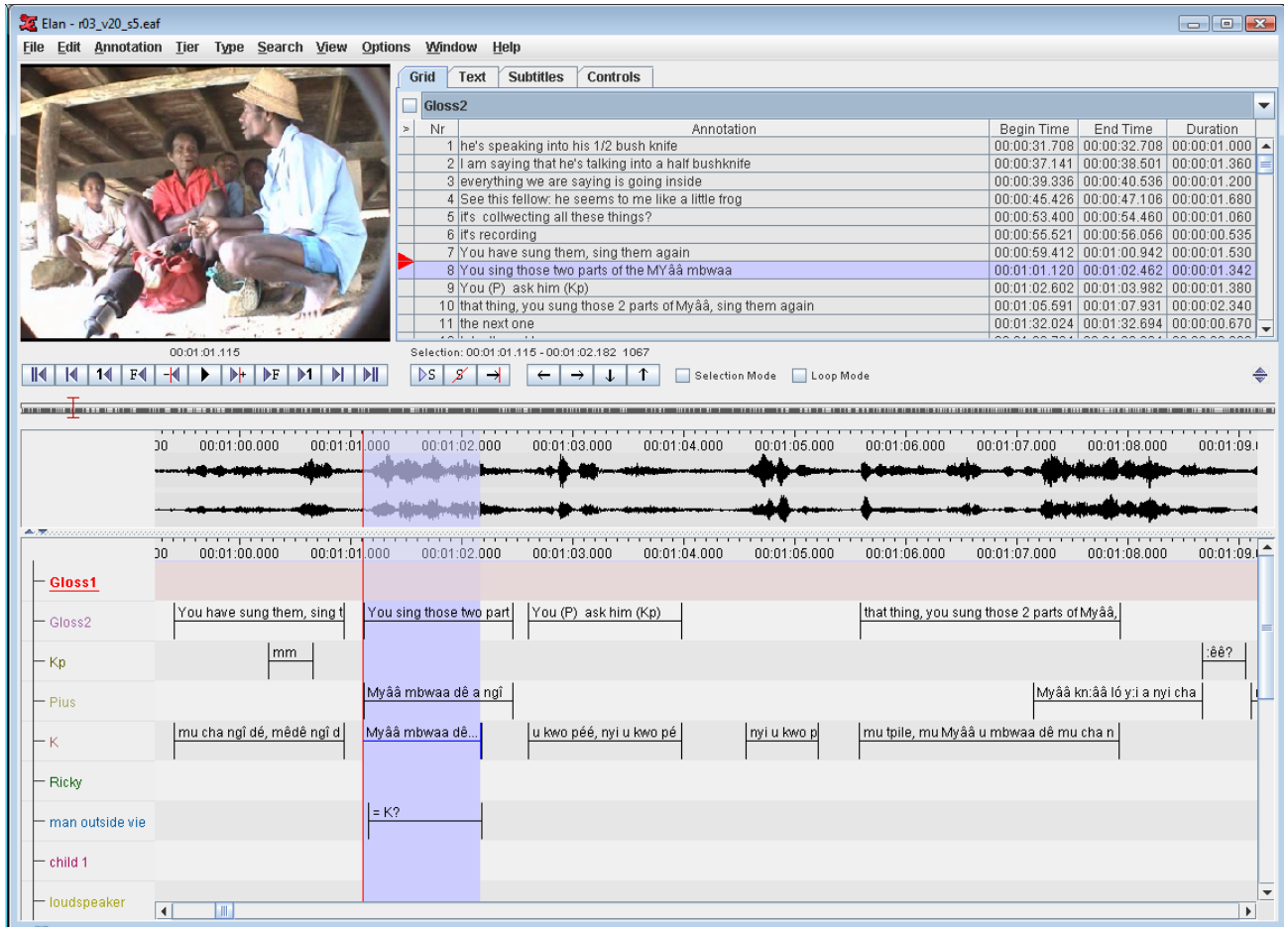
Developer: Michael Kipp, DFKI Saarbrücken

Tools (2): C-BAS



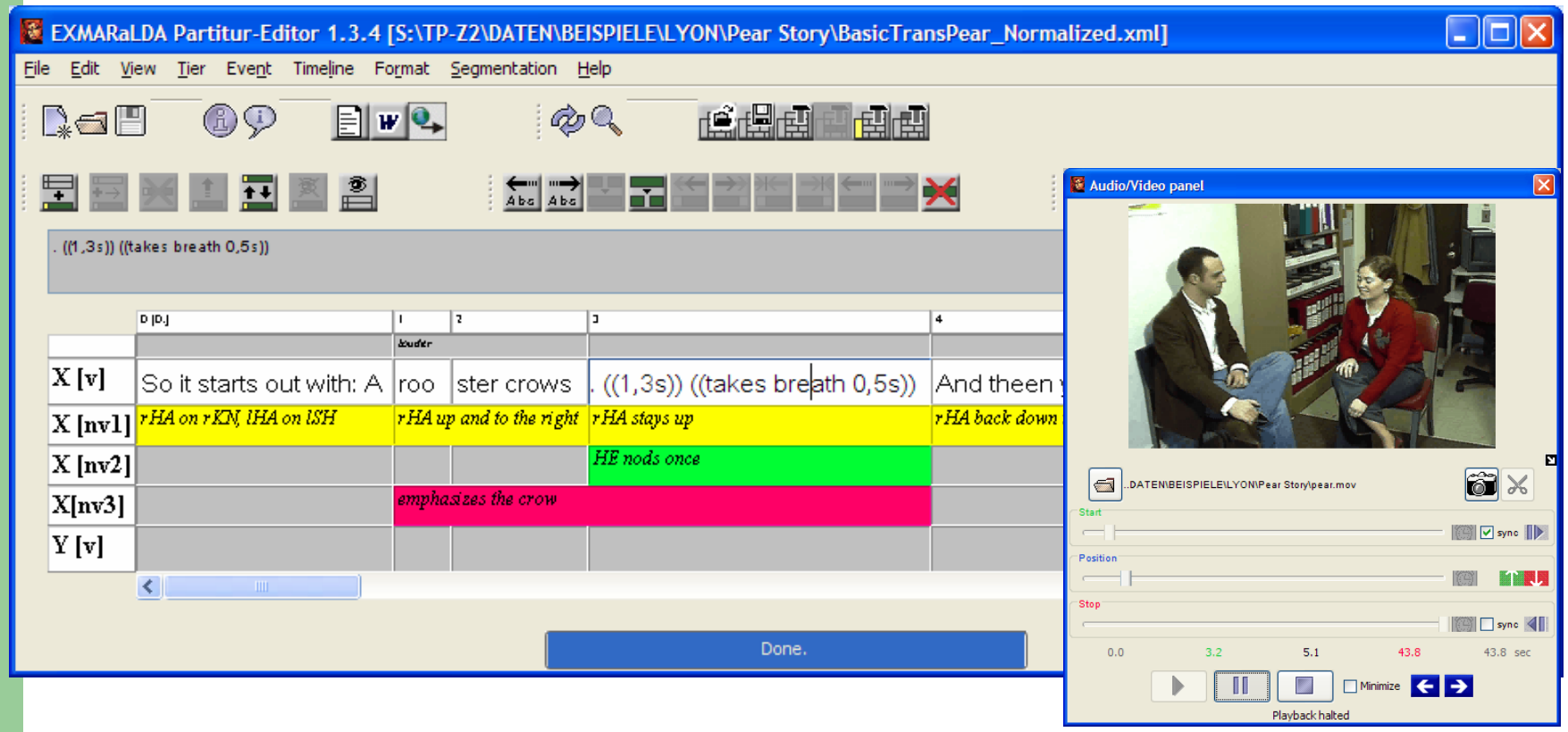
Developer: Kevin Moffit, University of Arizona

Tools (3): ELAN



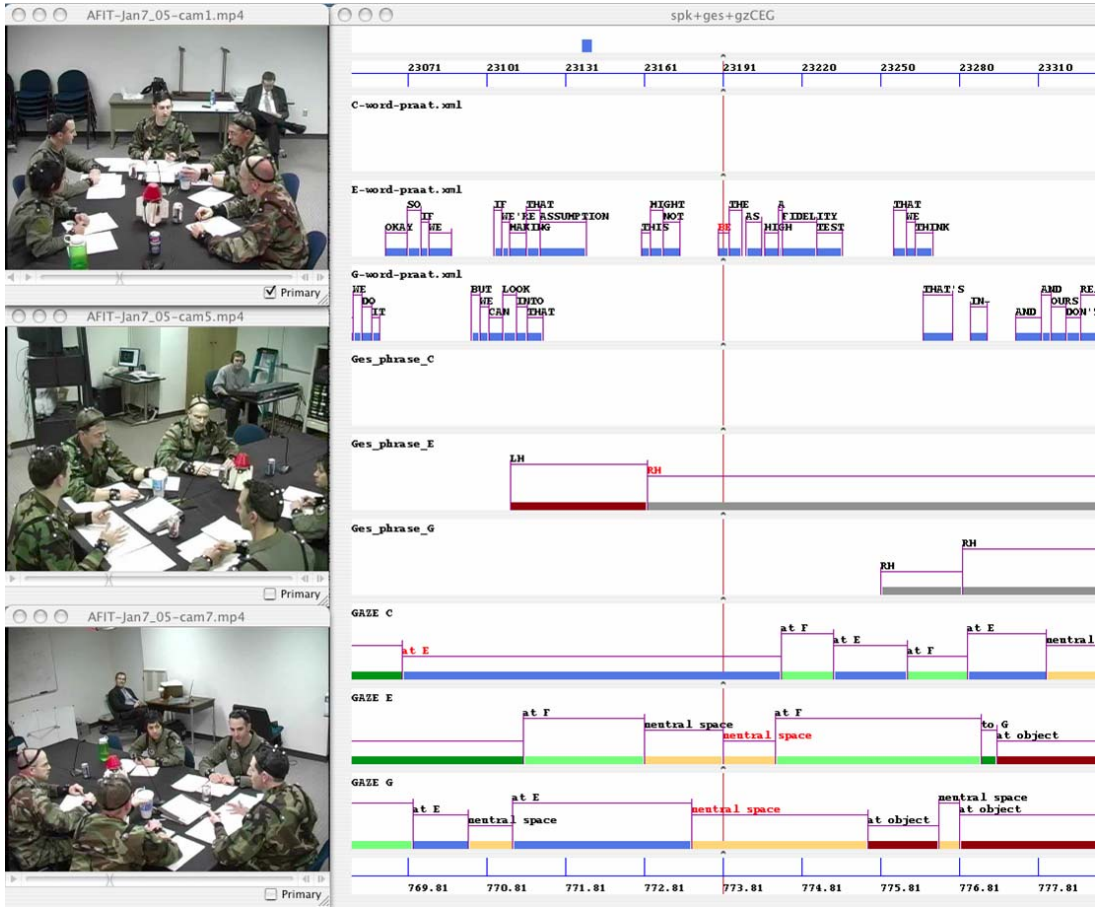
Developer: Han Sloetjes, MPI Nijmegen

Tools (4): EXMARaLDA Editor



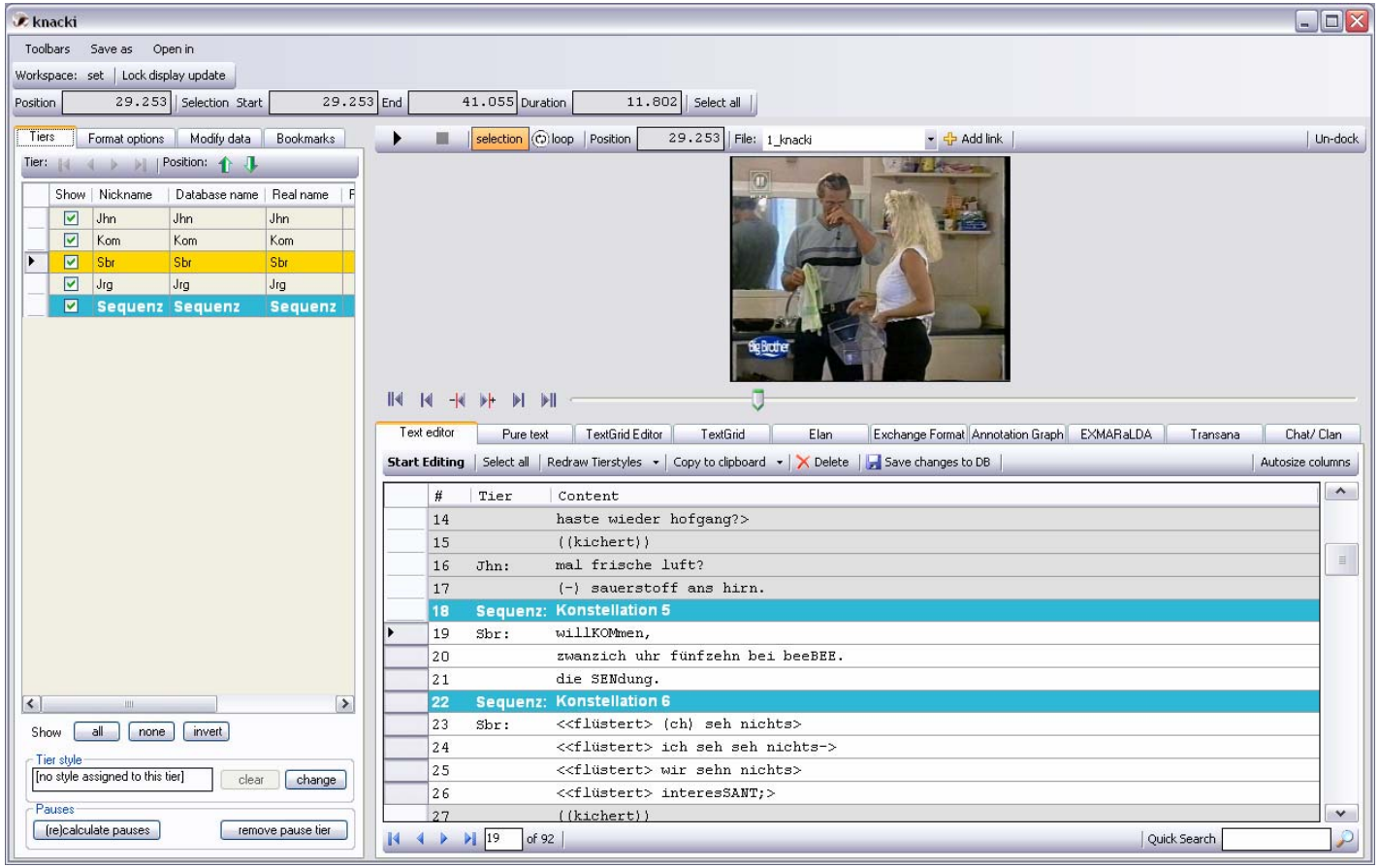
Developer: Thomas Schmidt, University of Hamburg

Tools (5): MacVisSTa



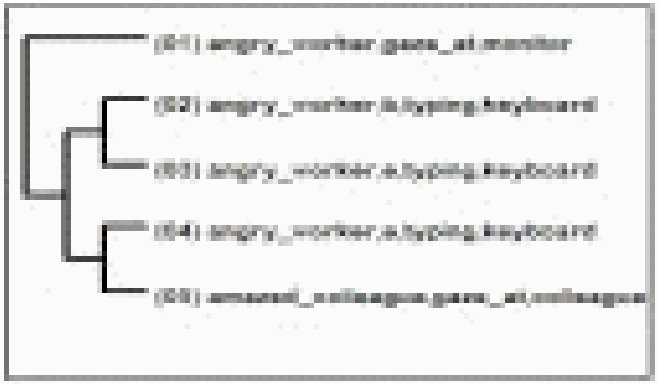
Developer: Travis Rose, Virginia Tech

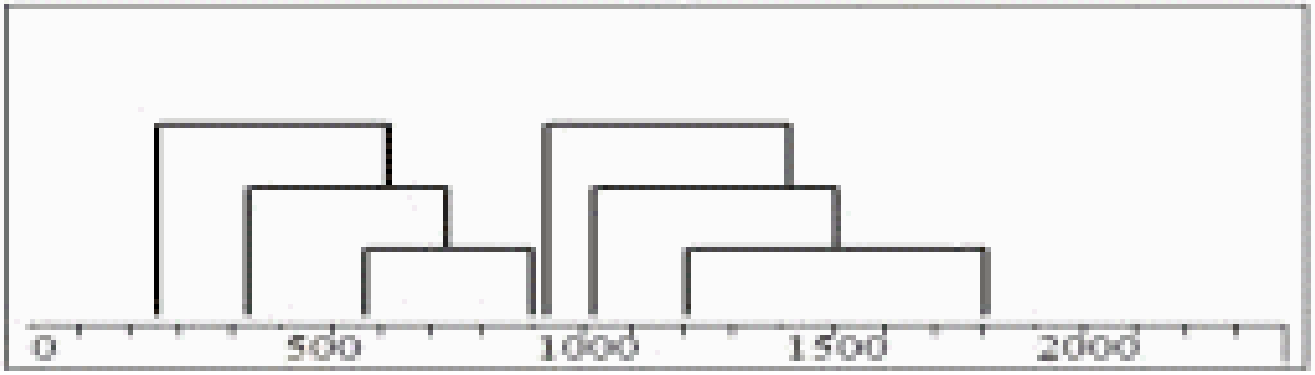
Tools (6): Transformer



Developer: Oliver Ehmer, University of Freiburg

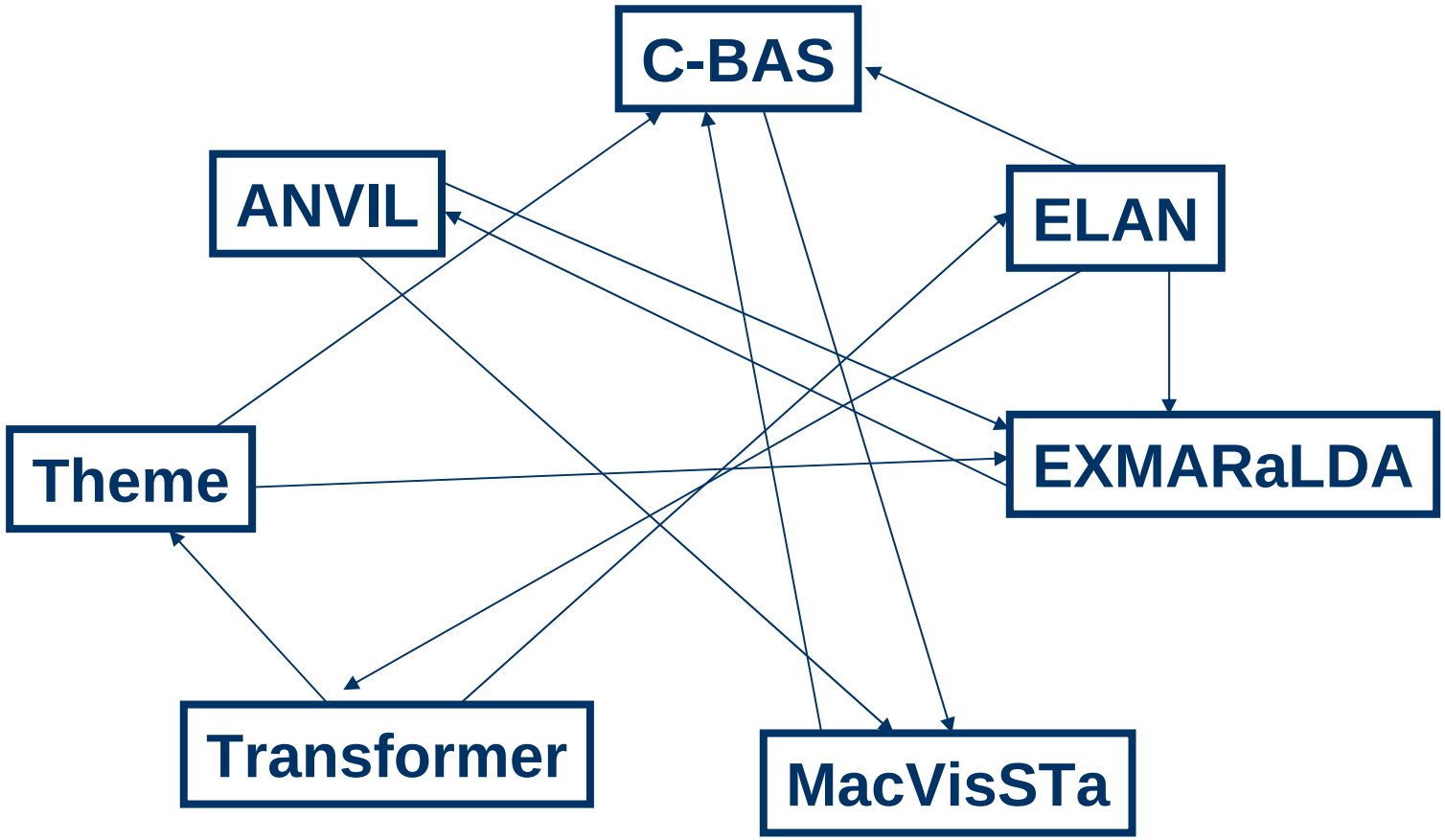
Tools (7): Theme



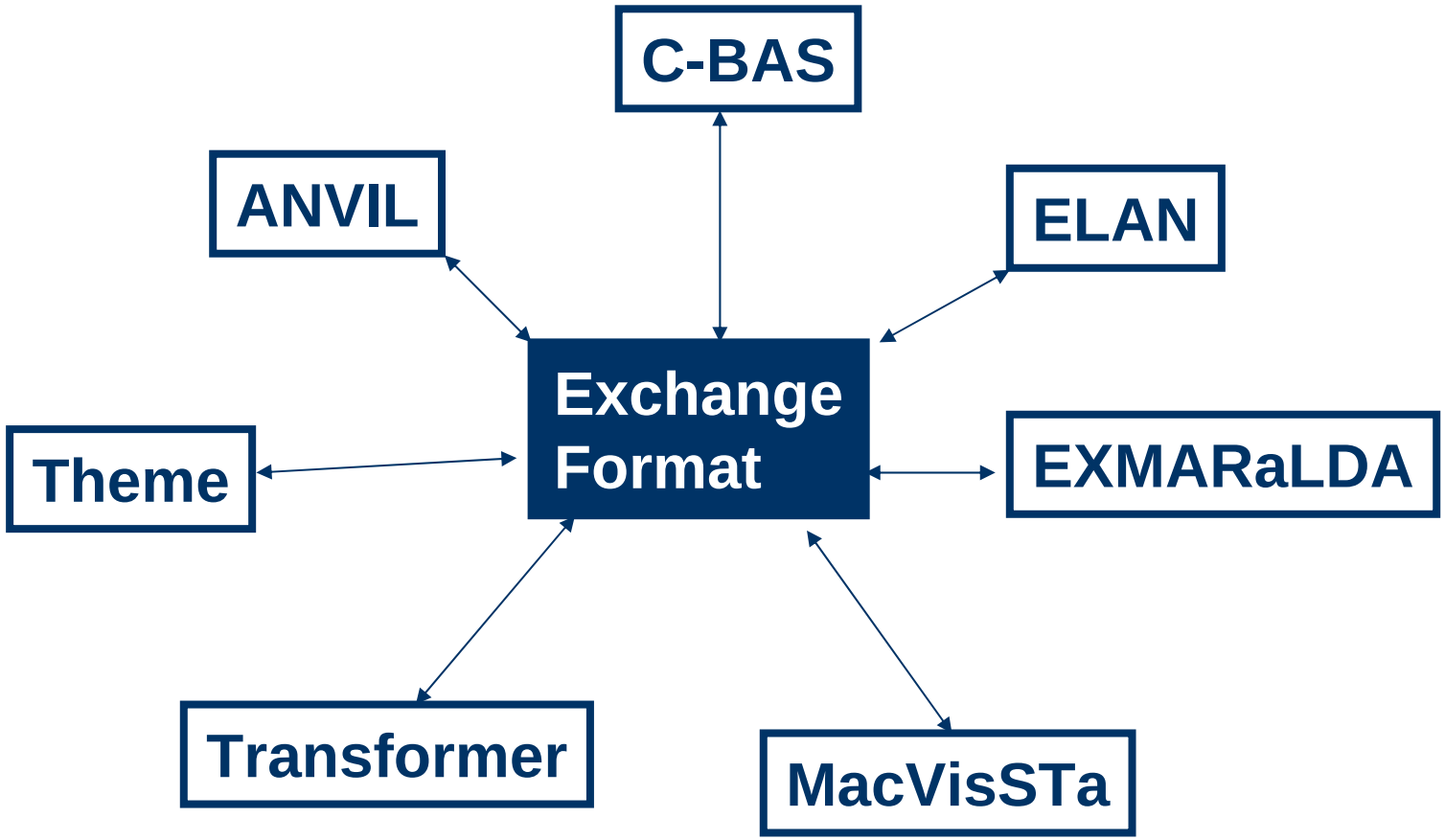


Developer: Magnus Magnusson, NOLDUS

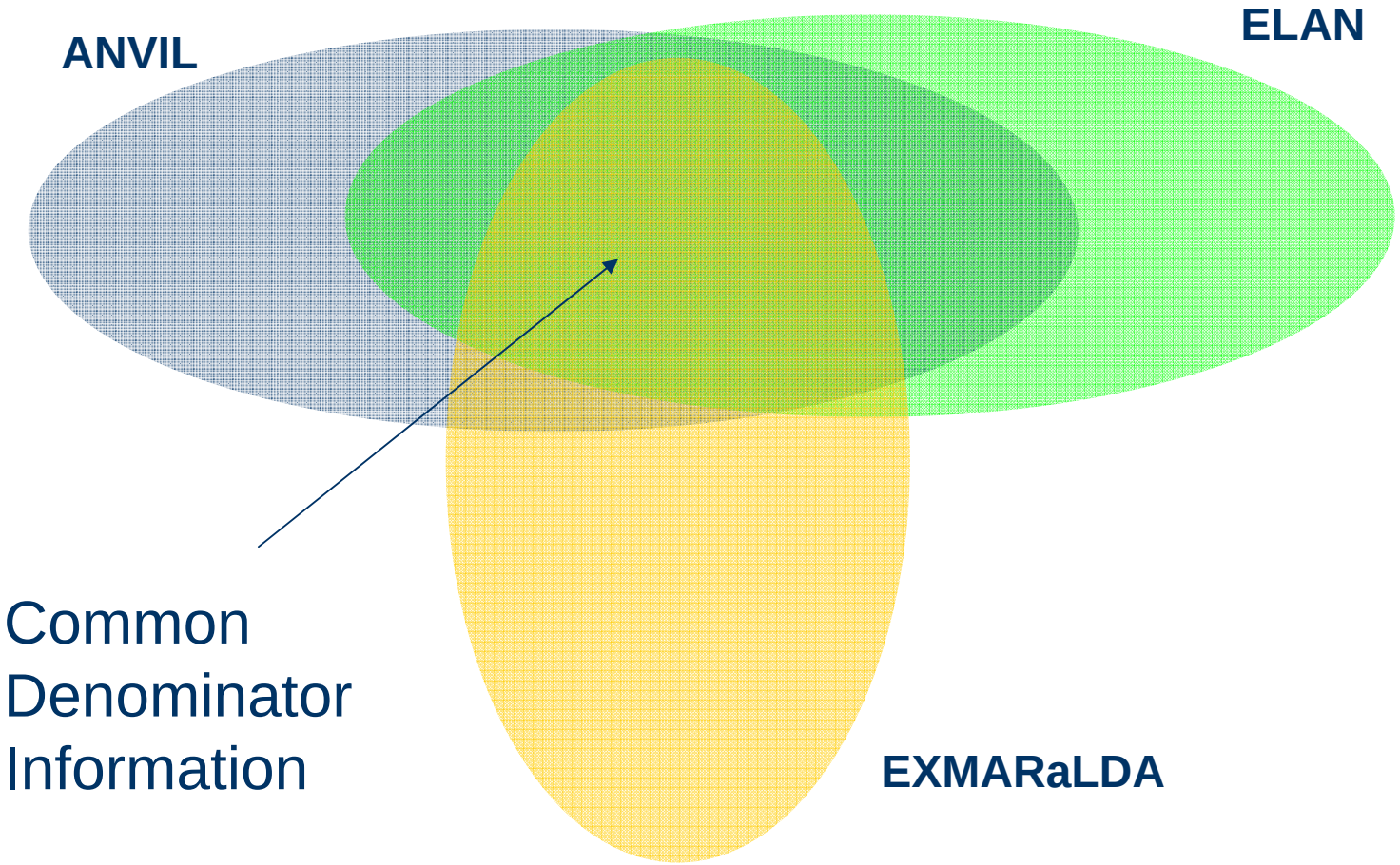
Interoperability



Interoperability



Data model comparison



Data model comparison

- Basic building blocks: Annotation tuples

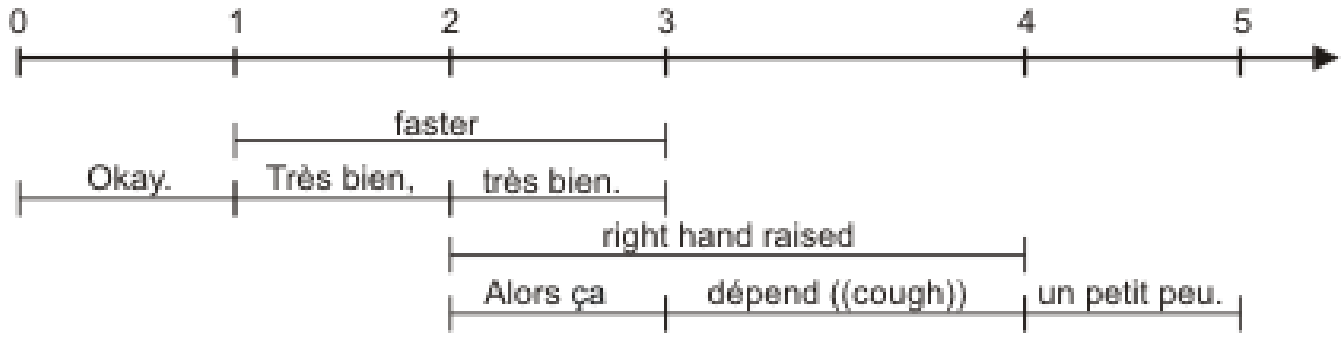
  `<start, end, label(s)>`

  → Annotation Graphs as a general framework

  → AG‘s XML format as the base format

- Differences:
  - General organisation of basic building blocks into larger structural units
  - Semantic specifications and constraints on structural units

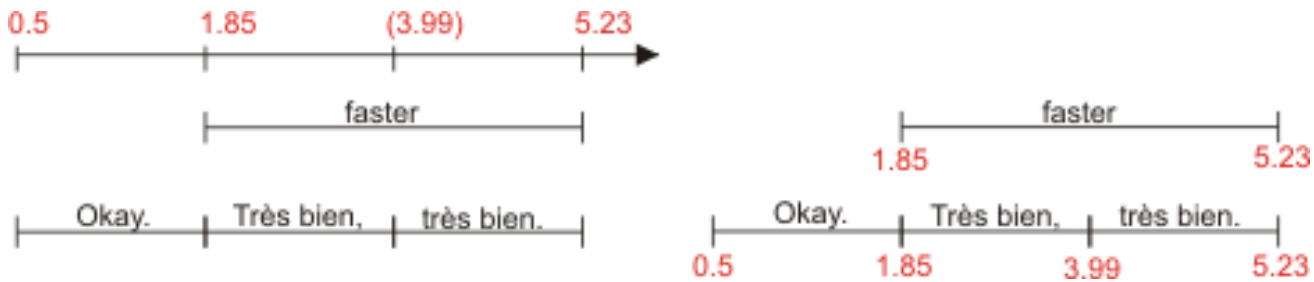
Tier-based vs. Non-tier-based



- Tiers = Partition of annotation tuples
- No temporal overlap within a tier
- In Anvil, ELAN, EXMARaLDA, Transformer

→ Construct partition from other information (e.g. categorisation of labels)

Implicit vs. explicit timeline



- Implicit timeline: annotation tuples refer directly to media times
- Explicit timeline: annotation tuples refer to points in a timeline which can refer to media times
  - Relative and absolute ordering of timepoints
  - Timepoints without timestamps possible

→ Interpolate timepoints without timestamps

→ Construct explicit timeline (identical timestamps?)

Tier specifications

- Tier names (all)
- Speaker assignment (ELAN, EXMARaLDA)
- Tier types:
  - ANVIL: primary, singleton, span
  - ELAN/Transformer: time subdivision, included in, symbolic subdivision, symbolic association
  - EXMARaLDA: transcription, description, annotation

Tier relations and constraints

- Parent/Child relations → tier hierarchy
  - explicit in Anvil and ELAN
  - implicit in EXMARaLDA
- Other constraints arising from tier typing
- Restrictions on label content
  - part of the tools‘ format?

Exchange Format

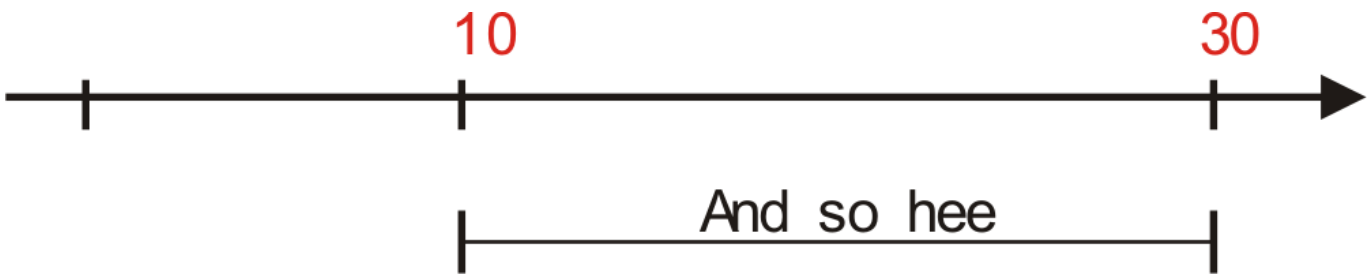
- Lossless exchange of common denominator information
- Uniformly encode all information beyond the common denominator

→ no lossless round-tripping, but...

→ ... all available information captured and...

→ ... lossless exchange in a chain of tools with increasingly complex data formats

Exchange Format




```
<Anchor id="T6" offset="10" unit="milliseconds"/>
<Anchor id="T7" offset="30" unit="milliseconds"/>
[…]
<Annotation type="TIE1" start="T6" end="T7">
    <Feature name="description">
        And so hee
    </Feature>
</Annotation>
```

Exchange Format

```
<MetadataElement name="Tier">
   <MetadataElement name="TierIdentifier">
   TIE1
   </MetadataElement>
 </MetadataElement>
 […]
<Annotation type="TIE1" start=" T6" end=" T7">
```

Tier definition:
Fixed metadata attribute
‚TierIdentifier‘

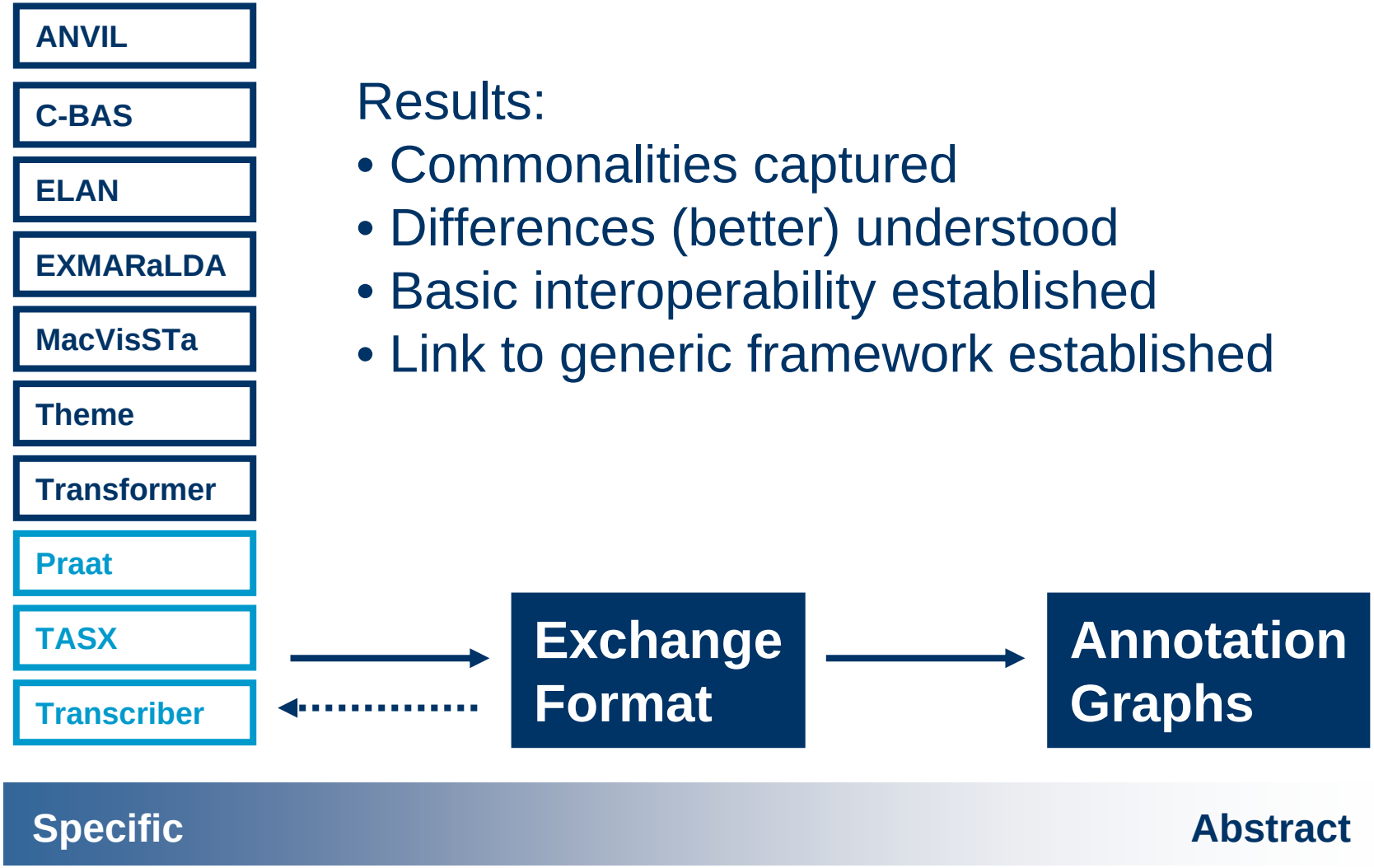
```
 <MetadataElement name="Tier">
    […]
    <MetadataElement name="TierAttribute">
        <MetadataElement name="Source">
            EXMARaLDA
        </MetadataElement>
        <MetadataElement name="Name">
            speaker
        </MetadataElement>
        <MetadataElement name="Value">
            SPK0
        </MetadataElement>
     </MetadataElement>
   </MetadataElement>
  […]
```

Tier properties:
Fixed metadata triple
‚Source‘
‚Name‘
‚Value‘

Implementation

- Import / Export routines
  - ANVIL, ELAN: AGLib (Java port)
  - EXMARaLDA: XSLT stylesheets
  - Theme: Perl
  - MacVisSTa: Python
  - Transformer: Visual Basic

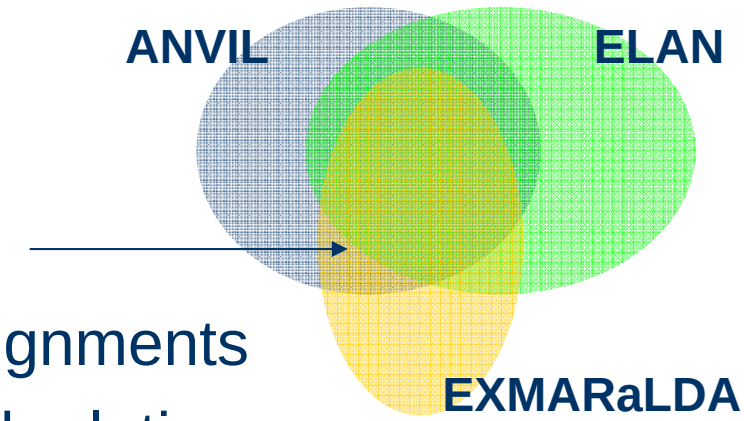
Conclusion



Results:

- Commonalities captured
- Differences (better) understood
- Basic interoperability established
- Link to generic framework established

Outlook



- Partial correspondences
  - Simple: e.g. speaker assignments
  - Complex: e.g. parent/child relations
- Modifying/Assimilating tools‘ formats
- „Process-based“ (as opposed to format-based) interoperability?