

# A Formal Framework For Interlinear Text

Kazuaki Maeda and Steven Bird  
Linguistic Data Consortium  
University of Pennsylvania  
{maeda , sb}@ldc . upenn . edu

# Outline

- Review of previous work
- Motivations
- Annotation Graphs (AGs)
- A framework for Interlinear Text
- Software

## Review of previous work

- T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X macros
- PTEXT
- IT and Shoebox – SIL standard format
- LACITO XML format
- Berkely Interlinear Text Collector (BITC)
- Kura

## Motivations

- An abstract data model which is independent of physical storage formats and display styles
- A model that can faithfully and efficiently represent all kinds of interlinear text commonly encountered, as well as all manipulations of interlinear text commonly performed.

## Annotation Graphs (AGs)

- Acyclic directed graphs (digraphs)
- Fielded records on the arcs
- Optional time references on the nodes

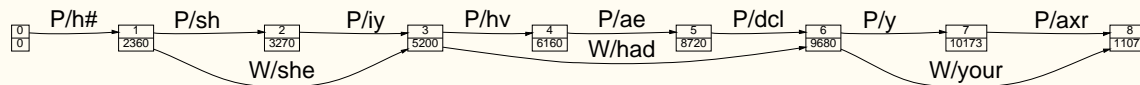


Figure 1: Annotation graph

## Definition of AG [Bird & Liberman '01]

**Definition 1** *An **annotation graph**  $G$  over a label set  $L$  and timelines  $\langle T_i, \leq_i \rangle$  is a 3-tuple  $\langle N, A, \tau \rangle$  consisting of a node set  $N$ , a collection of arcs  $A$  labeled with elements of  $L$ , and a time function  $\tau : N \rightarrow \bigcup T_i$ , which satisfies the following conditions:*

- 1.  $\langle N, A \rangle$  is a labeled acyclic digraph containing no nodes of degree zero;*
- 2. for any path from node  $n_1$  to  $n_2$  in  $A$ , if  $\tau(n_1)$  and  $\tau(n_2)$  are defined, then there is a timeline  $i$  such that  $\tau(n_1) \leq_i \tau(n_2)$ .*

## Subgraphs of AGs

**Definition 2** An AG  $\langle N', A', \tau' \rangle$  is a **subgraph** of an AG  $\langle N, A, \tau \rangle$  iff  $A' \subseteq A$ ; and  $N'$  and  $\tau'$  are the restriction of  $N$  and  $\tau$  to just those nodes used by  $A'$ . If  $G'$  is a subgraph of  $G$  we write  $G' \subseteq G$ .

## **Advantages of AGs**

- An efficient and expressive data model for annotating time-series data.

## Advantages of AGs

- An efficient and expressive data model for annotating time-series data.
- Efficient query (via a direct representation as a relation table) [Bird et al., 2000].

## Advantages of AGs

- An efficient and expressive data model for annotating time-series data.
- Efficient query (via a direct representation as a relation table) [Bird et al., 2000].
- Incomplete information can be represented naturally.

## Applying AG to Interlinear Text

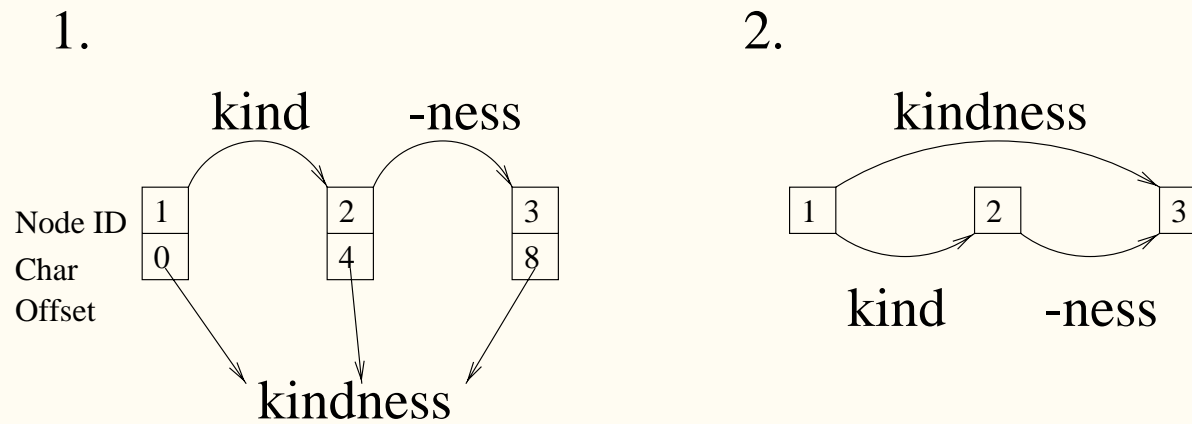
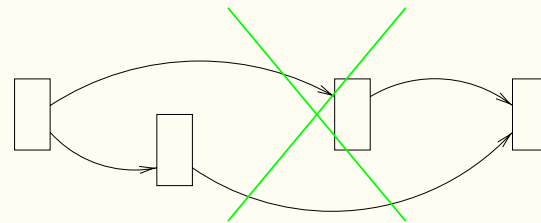
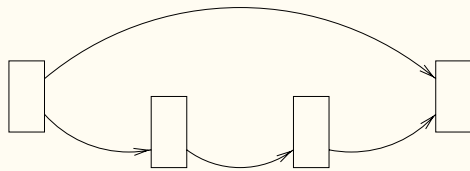
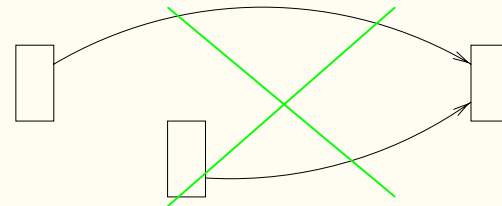
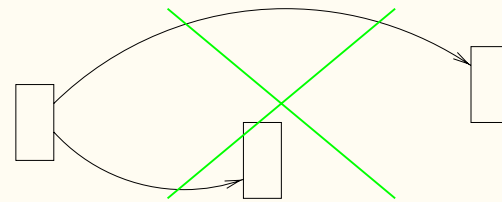
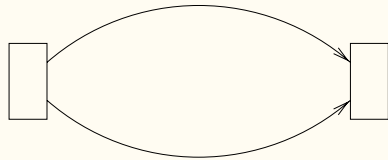


Figure 2: Two ways of expressing interlinear texts with AGs

# Structural limitations in Interlinear Text



## **A formal framework for Interlinear Text based on AGs**

- Conditions for interlinear text — the full expressive power of AGs is not needed.
- For the development of special purpose GUI for interlinear text

## Grouping of types

Types: Words, Word-glosses, Morphemes, Morpheme-glosses, Phonemes, etc.

**Condition 1** *Every type belongs to exactly one group.*

**Definition 3** *A group  $n$  annotation subgraph  $G_n \langle N_n, A_n, \tau_n \rangle$  for interlinear text is a subgraph of an annotation graph  $G \langle N, A, \tau \rangle$ , where  $A_n$  consists of all group  $n$  arcs in  $A$ .*

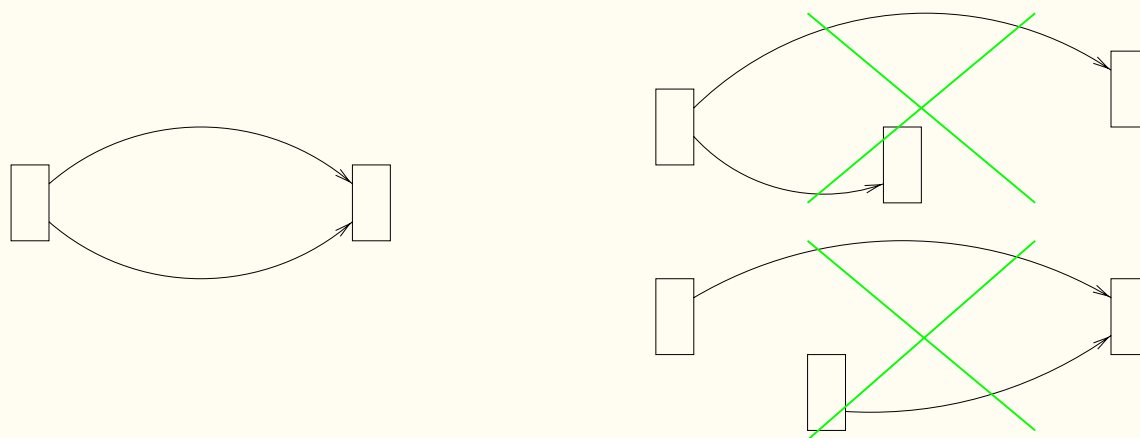


Figure 3: Structural limitation

**Condition 2** *A group  $n$  annotation subgraph  $G_n$  for interlinear texts must satisfy the following condition; every arc leaving a node  $n_1$  in  $G_n$  must end with the same node  $n_2$  in  $G_n$ , and every arc coming into a node  $n_2$  in  $G_n$  must start with the same node  $n_1$  in  $G_n$ .*

## Containment of groups

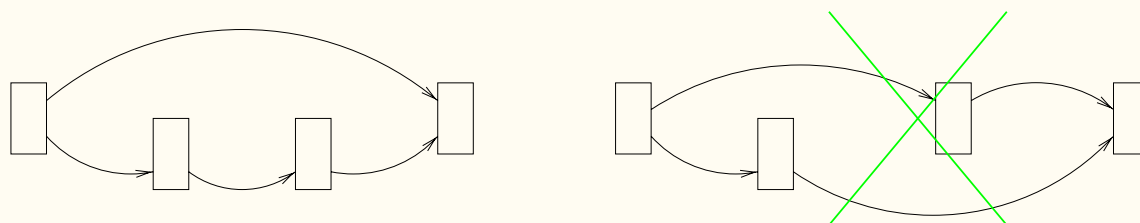
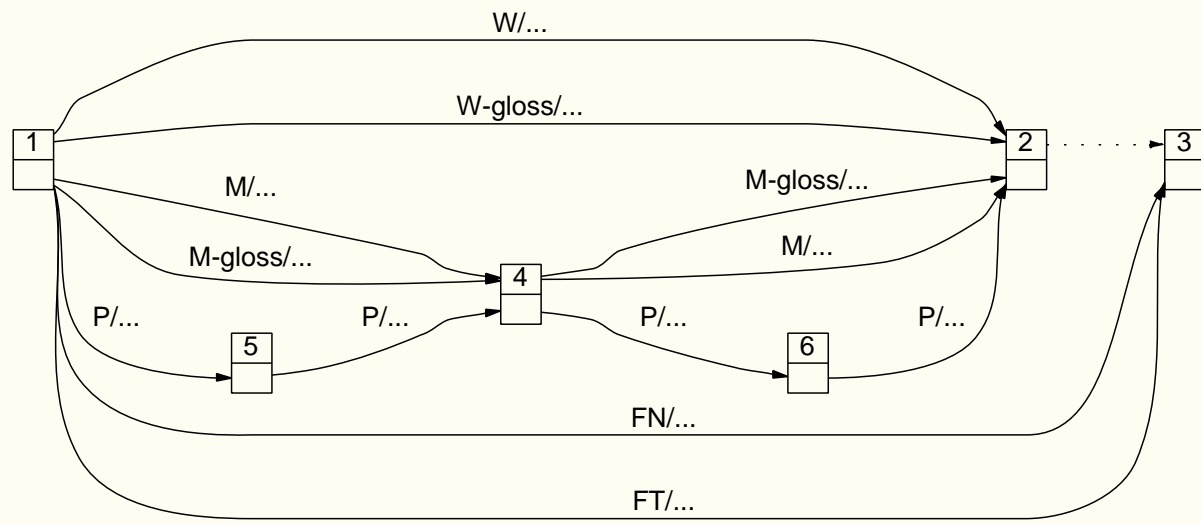
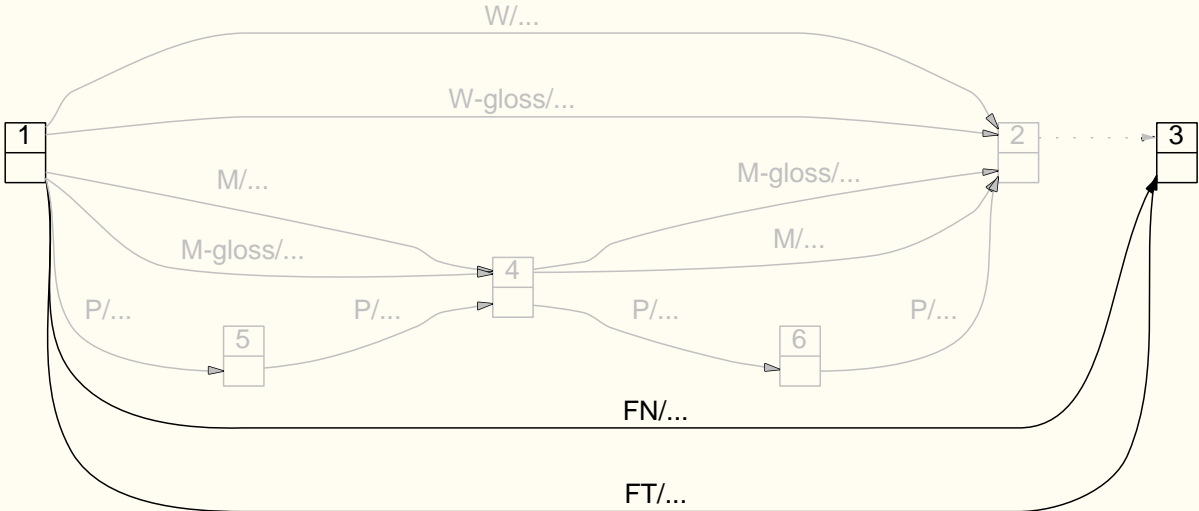


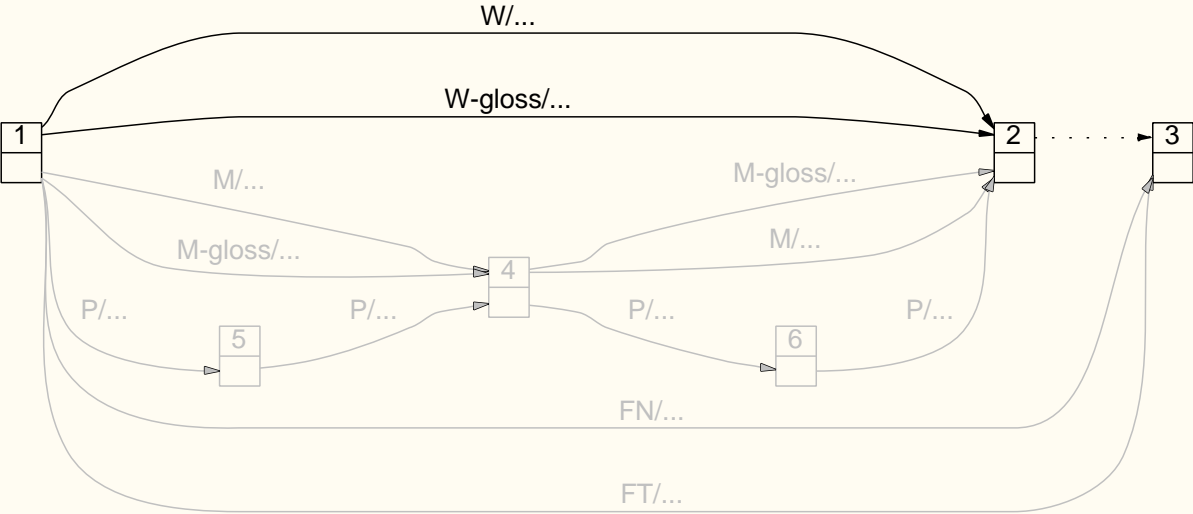
Figure 4: Containment

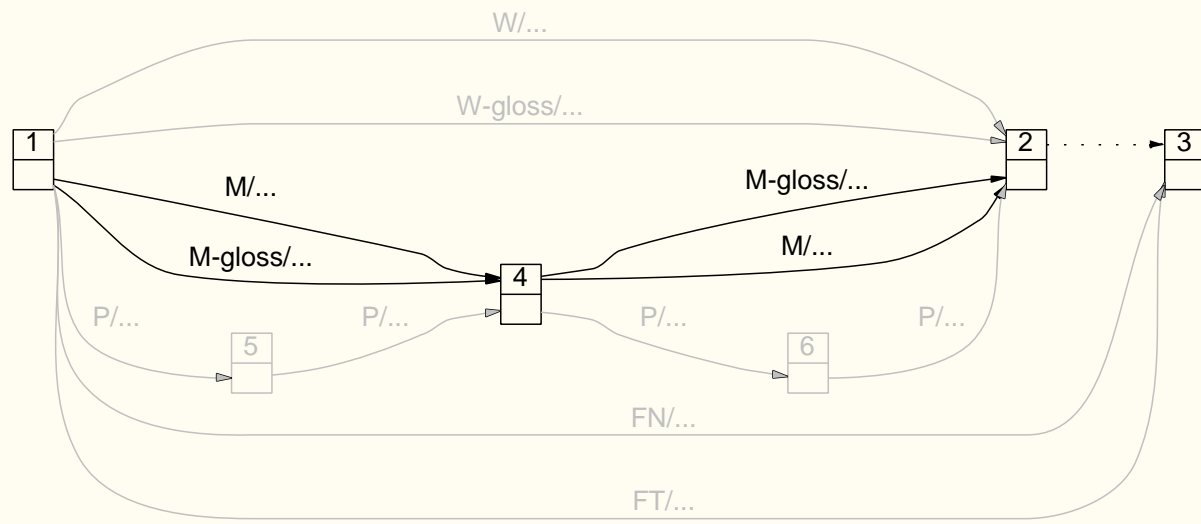
**Definition 4** A group  $n$  annotation subgraph  $G_n \langle N_n, A_n, \tau_n \rangle$  of  $G \langle N, A, \tau \rangle$  for interlinear texts **contains** a group  $m$  annotation subgraphs  $G_m \langle N_m, A_m, \tau_m \rangle$  of  $G$  iff  $N_n \subseteq N_m$  and  $N_n \neq N_m$ .

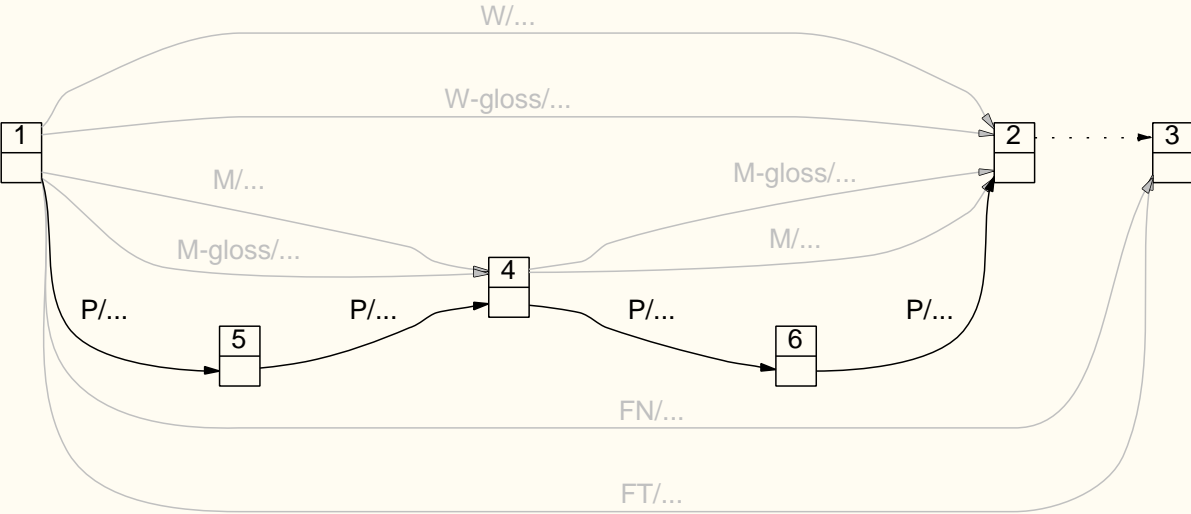
**Condition 3** For any group  $m$  annotation subgraph  $G_m$  of annotation graph  $G$ , if there exists a group  $n$  annotation subgraph  $G_n$  of  $G$  that contains  $G_m$ , then group  $n$  must contain group  $m$ .











## Example in Mawu

SW: Yala o Bulama minn00n niyE  
 M: Yala o yE Bulama min l00n, n yE a ye  
 MG: lion the AUX Ibrahim WH eat, I AUX him saw  
 FT: ``I saw Ibrahim, whom a lion ate.``  
 FN: Basic rel. clause with obj. as relative element.

- Group S: FT, FN
- Group W: SW
- Group M: M, MG
  
- Group S *contains* group W.
- Group W *contains* group M.

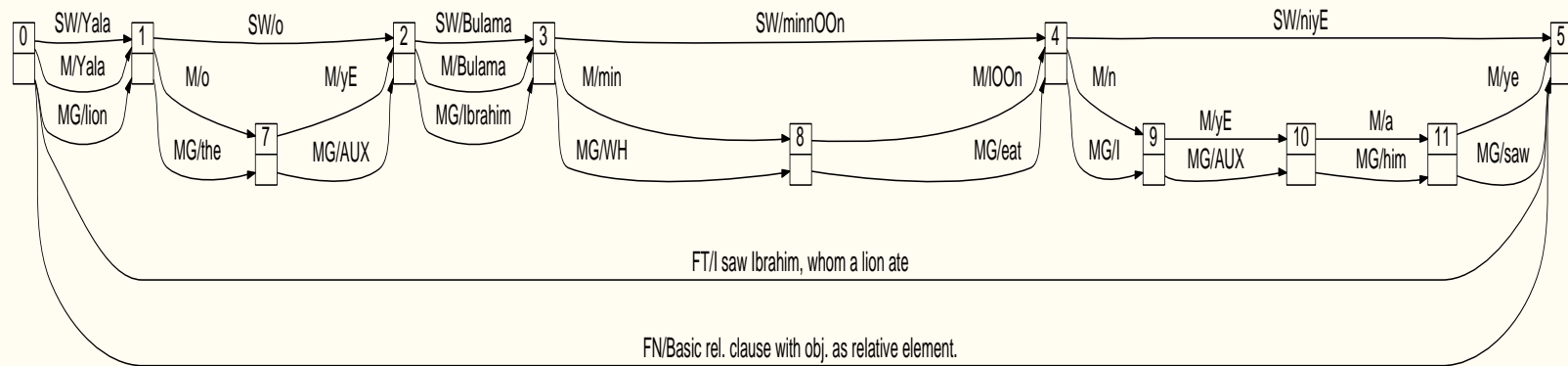


Figure 5: Mawu example

# Software

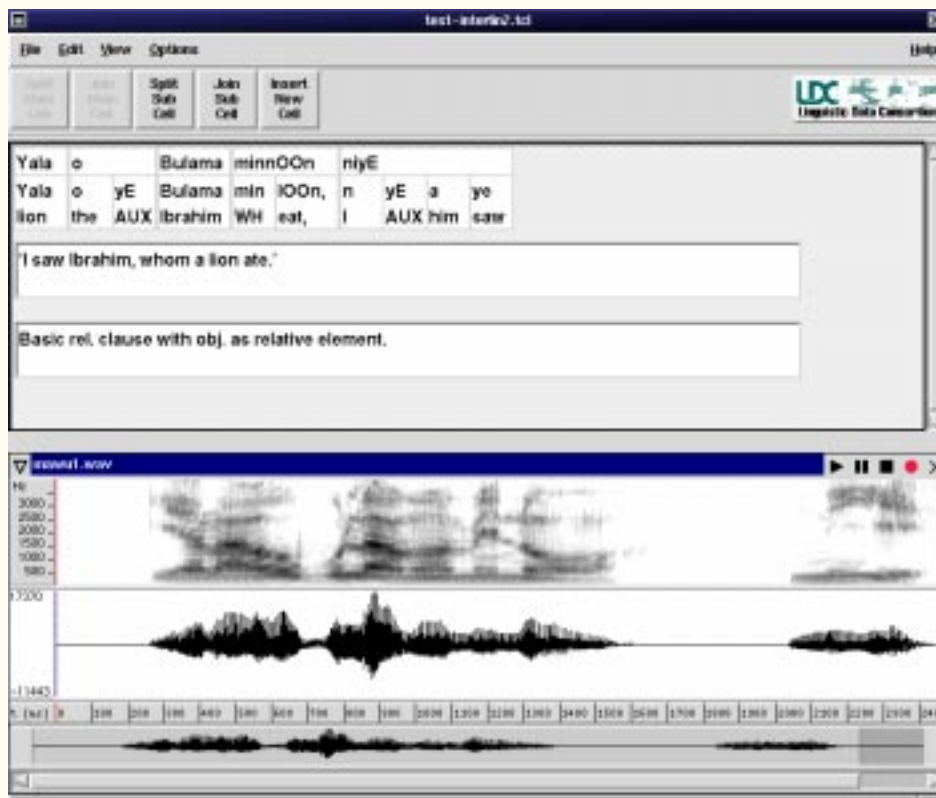


Figure 6: Prototype Interlinear Text Editor

# Software

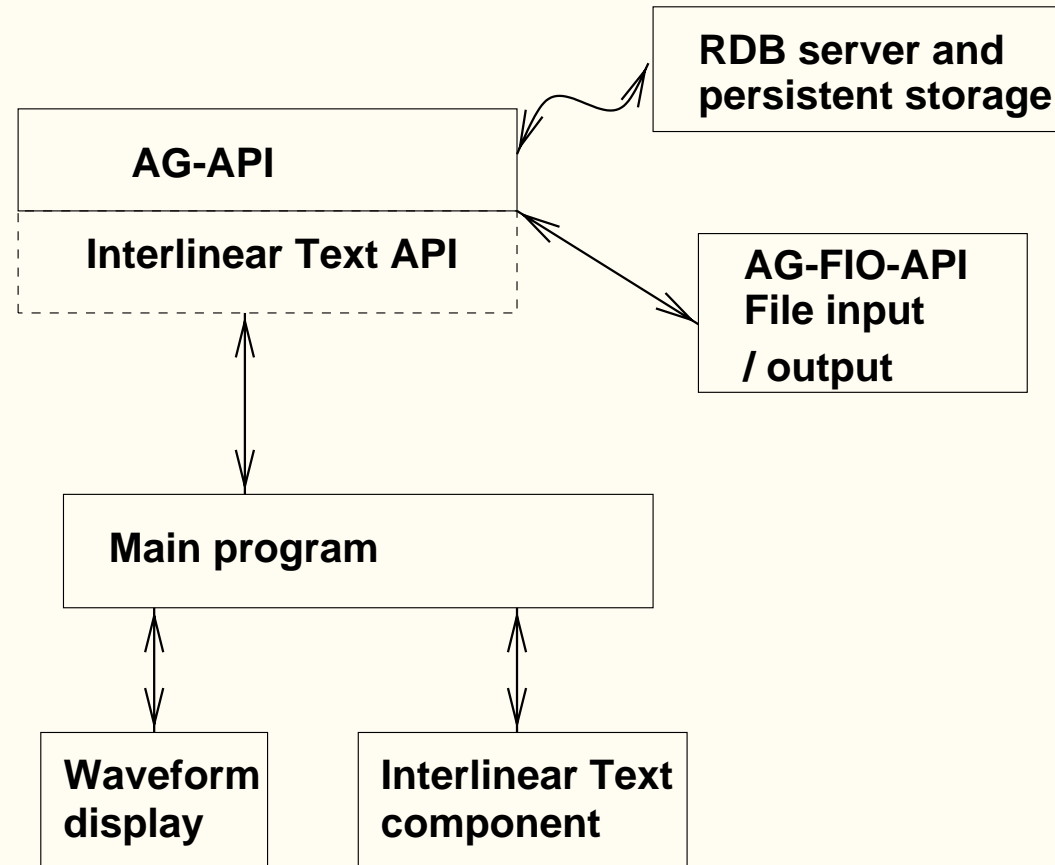


Figure 7: Architecture based on AG (Interlinear Text) API

Yala	o		Bulama	minnOOn		niyE			
Yala	o	yE	Bulama	min	IOOn,	n	yE	a	ye
lion	the	AUX	Ibrahim	WH	eat,	I	AUX	him	saw
<p>'I saw Ibrahim, whom a lion ate.'</p>									
<p>Basic rel. clause with obj. as relative element.</p>									

Figure 8: Interlinear Text Component

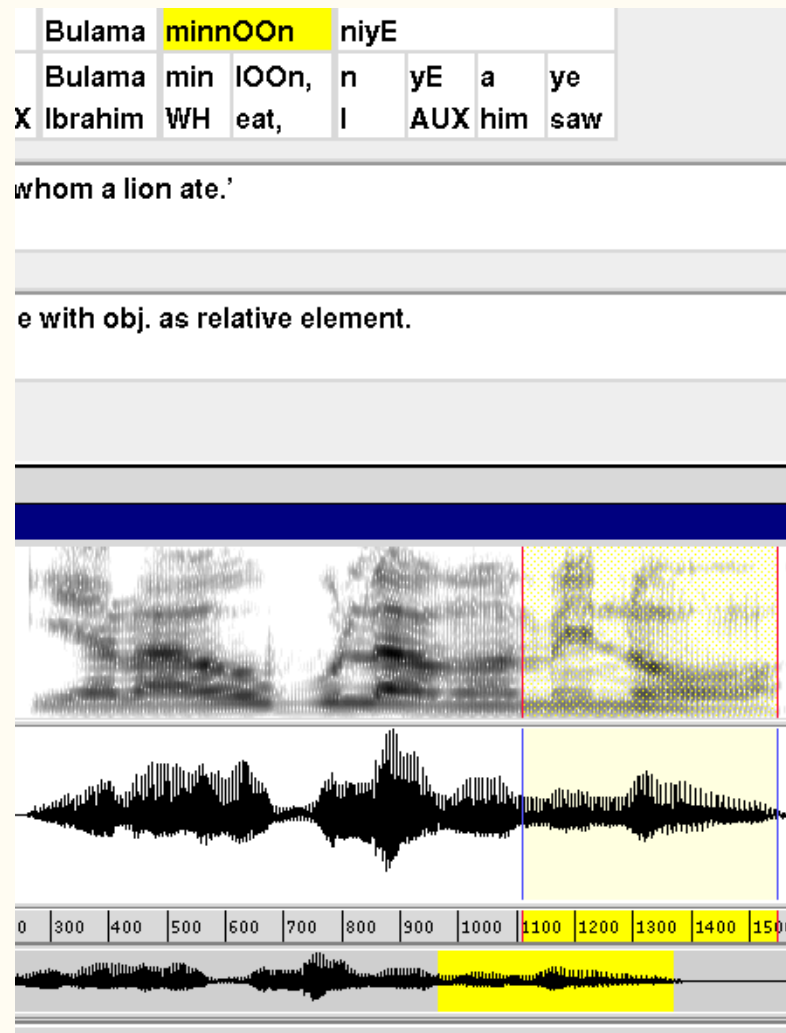


Figure 9: Aligning text with speech

## Future Plans

- File I/O and Various Formats
- Presentation based on XSL stylesheets
- Lexicon and Morphological Analyzer
- Interlinear text API and Component Design
- Collaborative Editing
- Generalized Models

## Summary

- A general framework for interlinear text using the AG model
- An interlinear text editor using this framework
- Plans for an API and reusable components