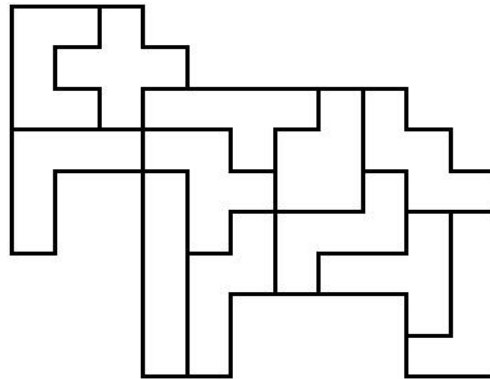
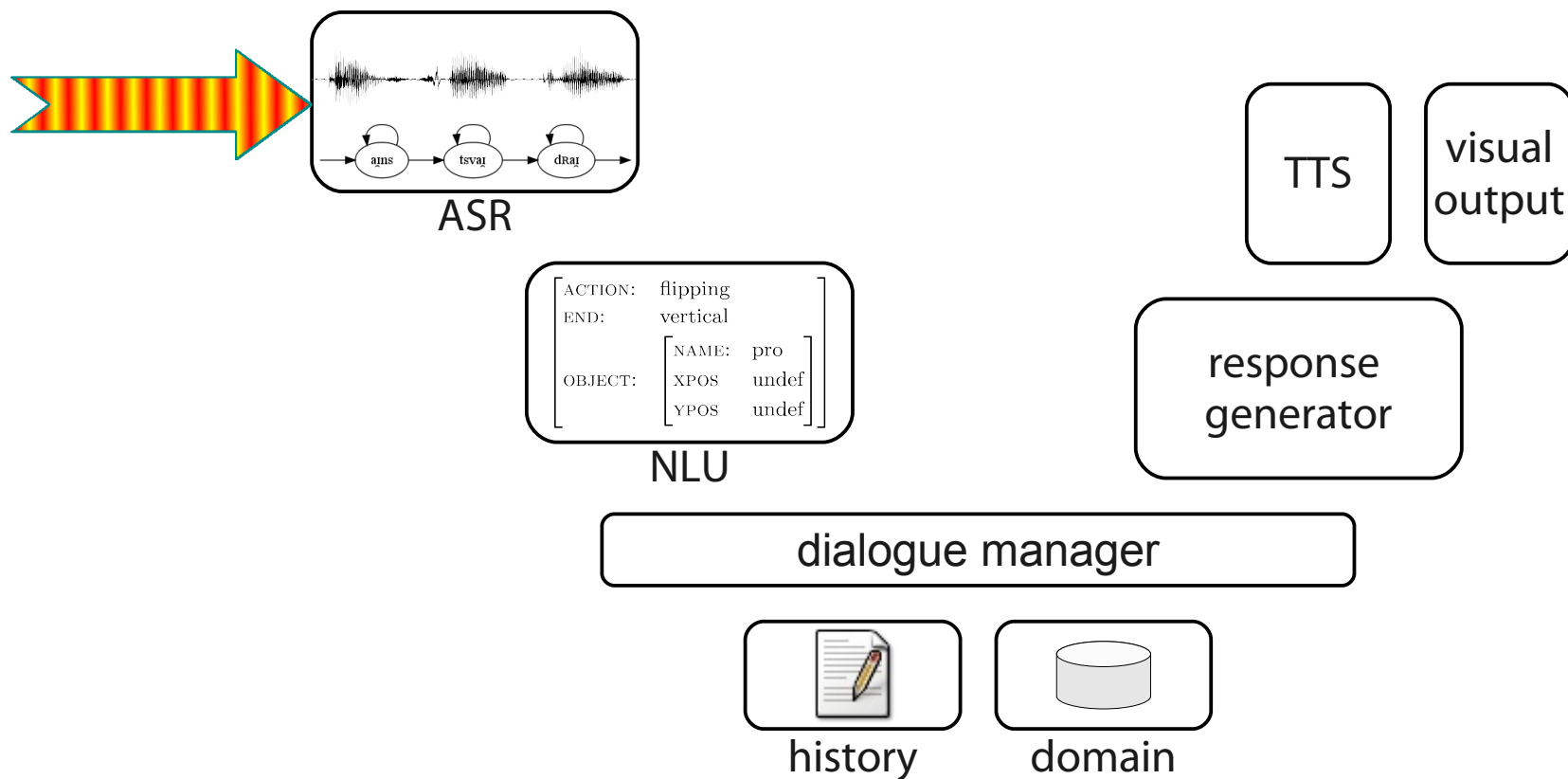


Incremental ASR, NLU and Dialogue Management in the Potsdam INPRO P2 System

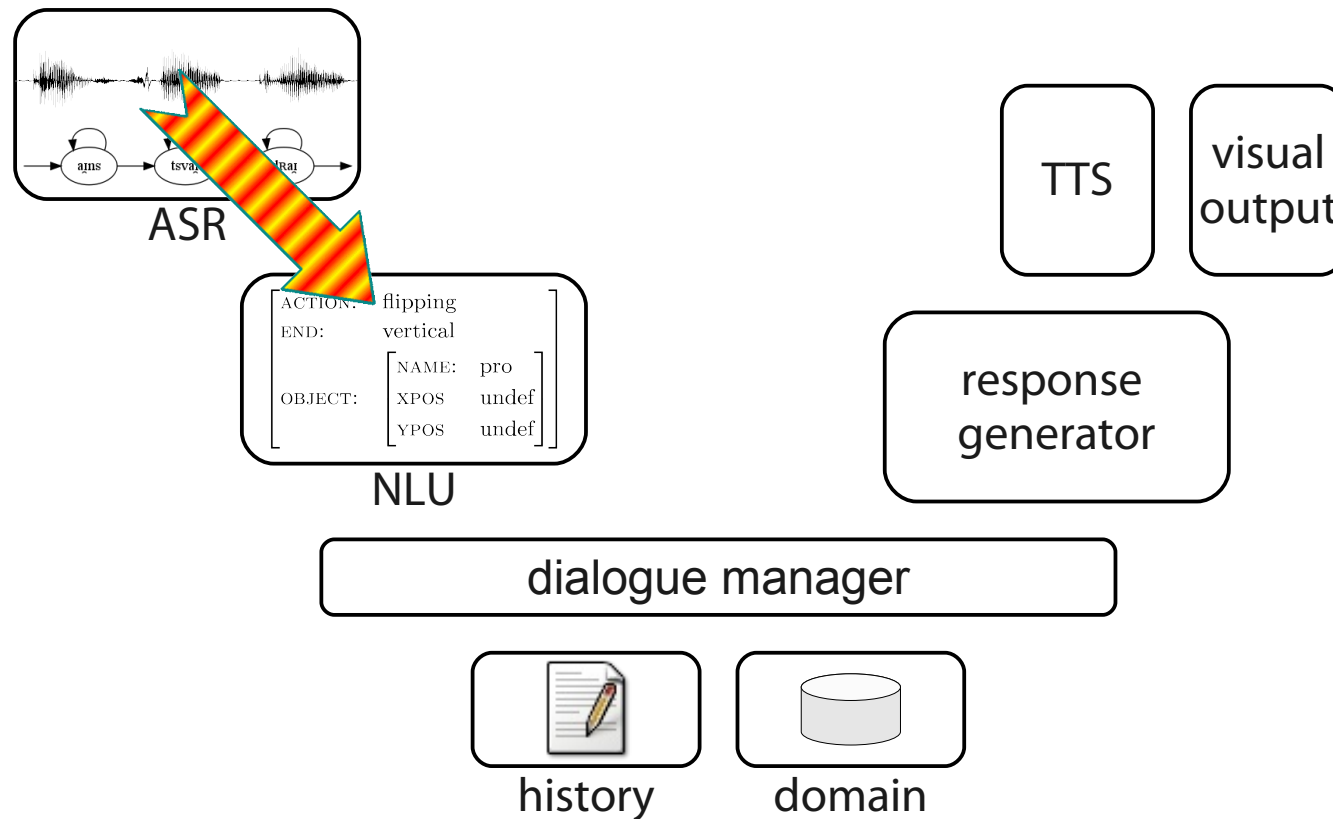


Timo Baumann, Michaela Atterer, Okko Buss
IVI Workshop, Bielefeld, 2009-06-08

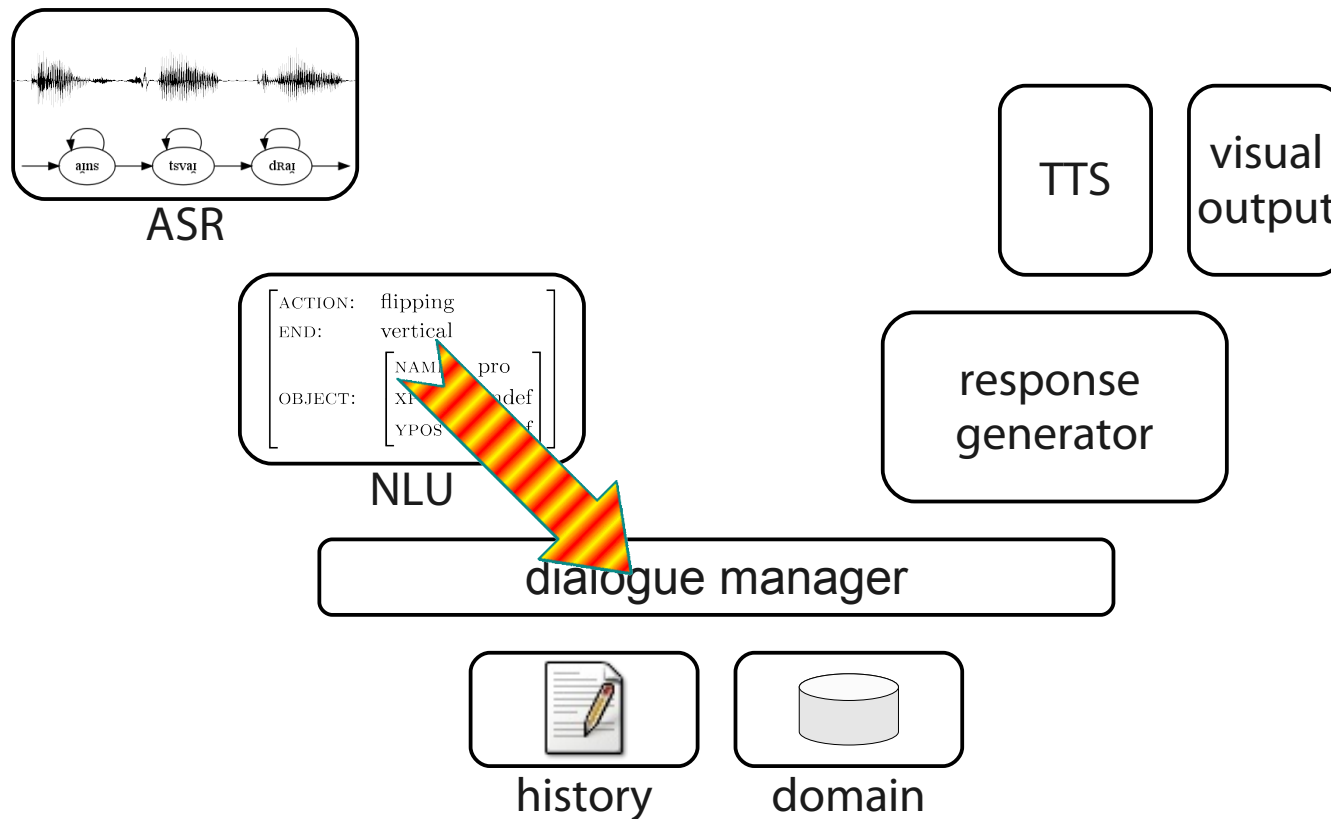
Context: Spoken Dialogue Systems



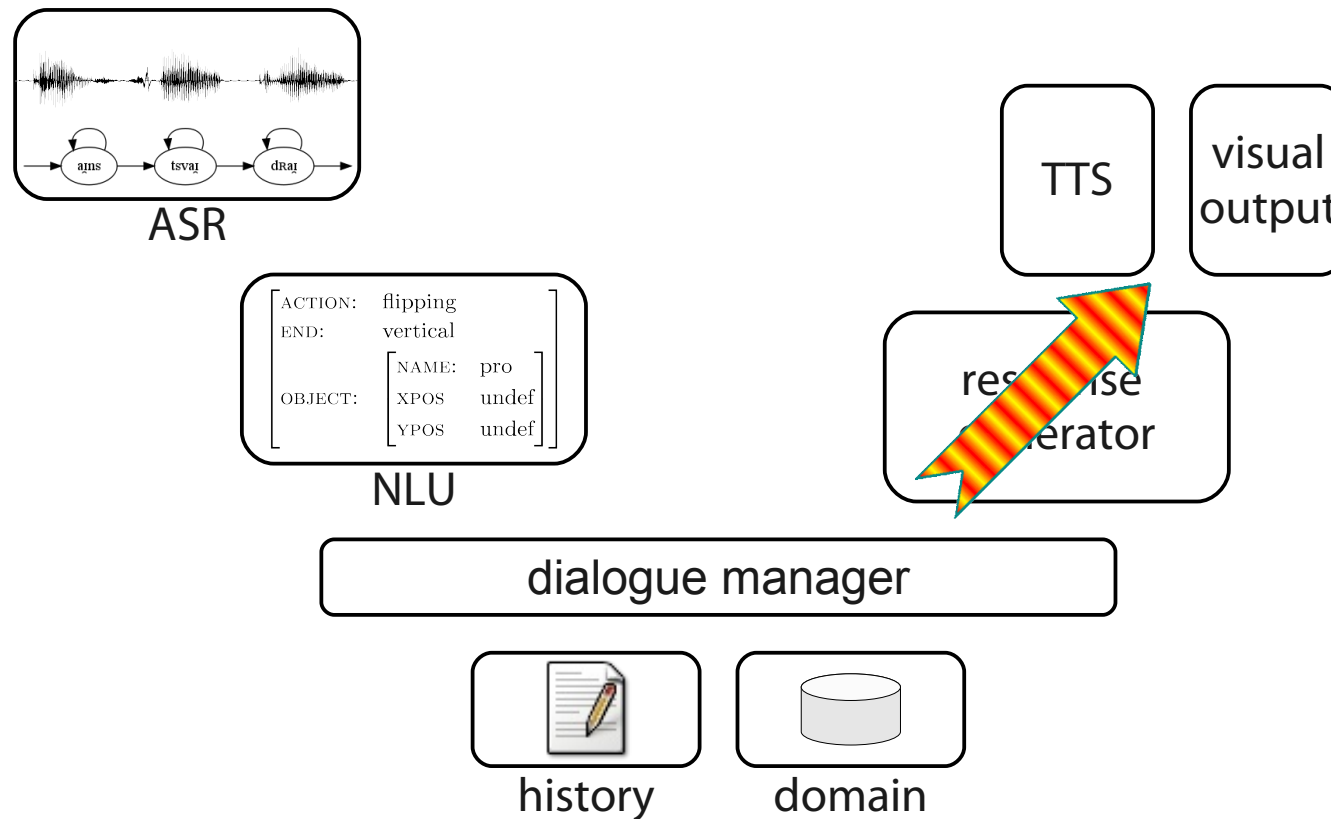
Context: Spoken Dialogue Systems



Context: Spoken Dialogue Systems

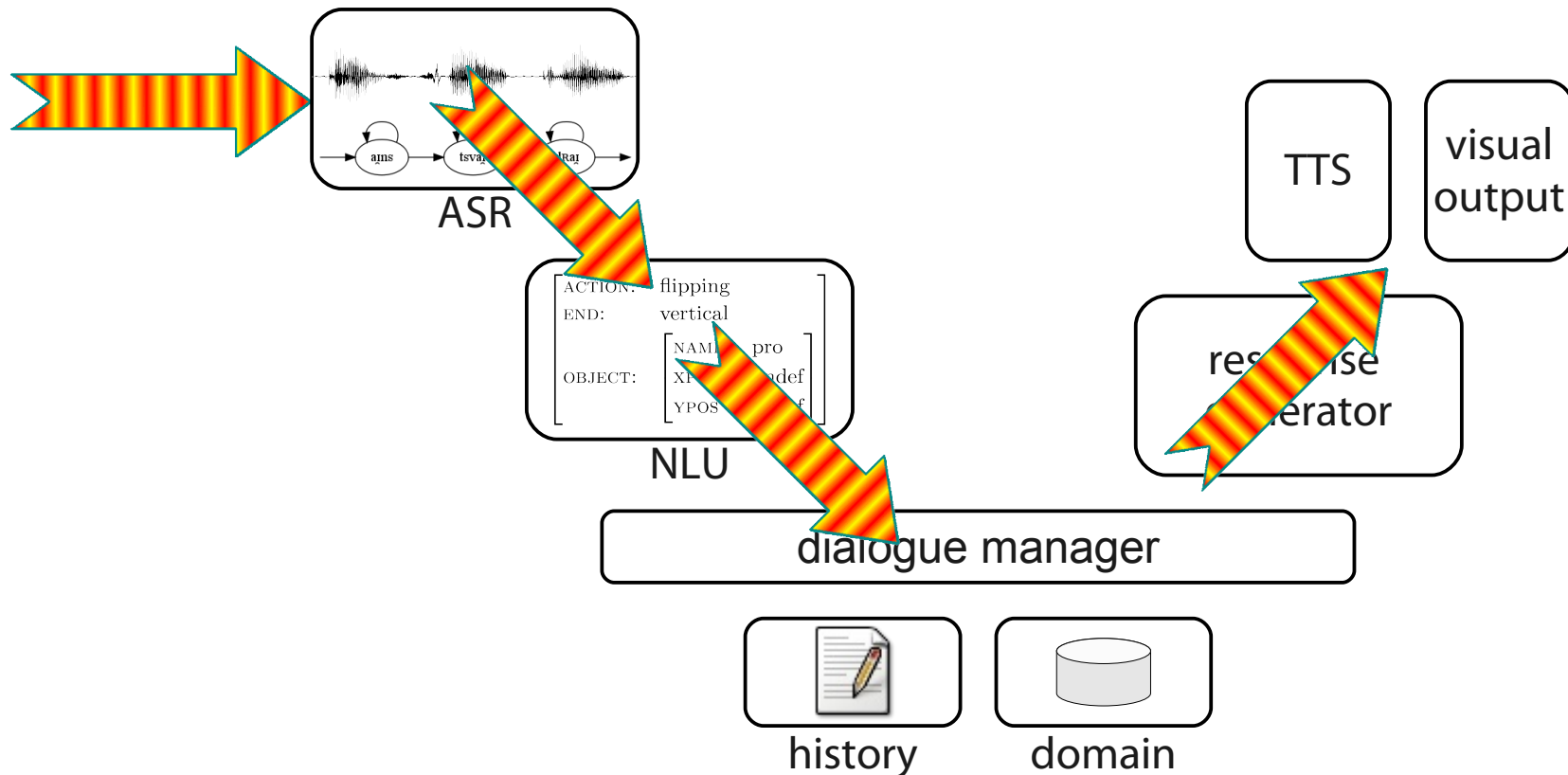


Context: Spoken Dialogue Systems



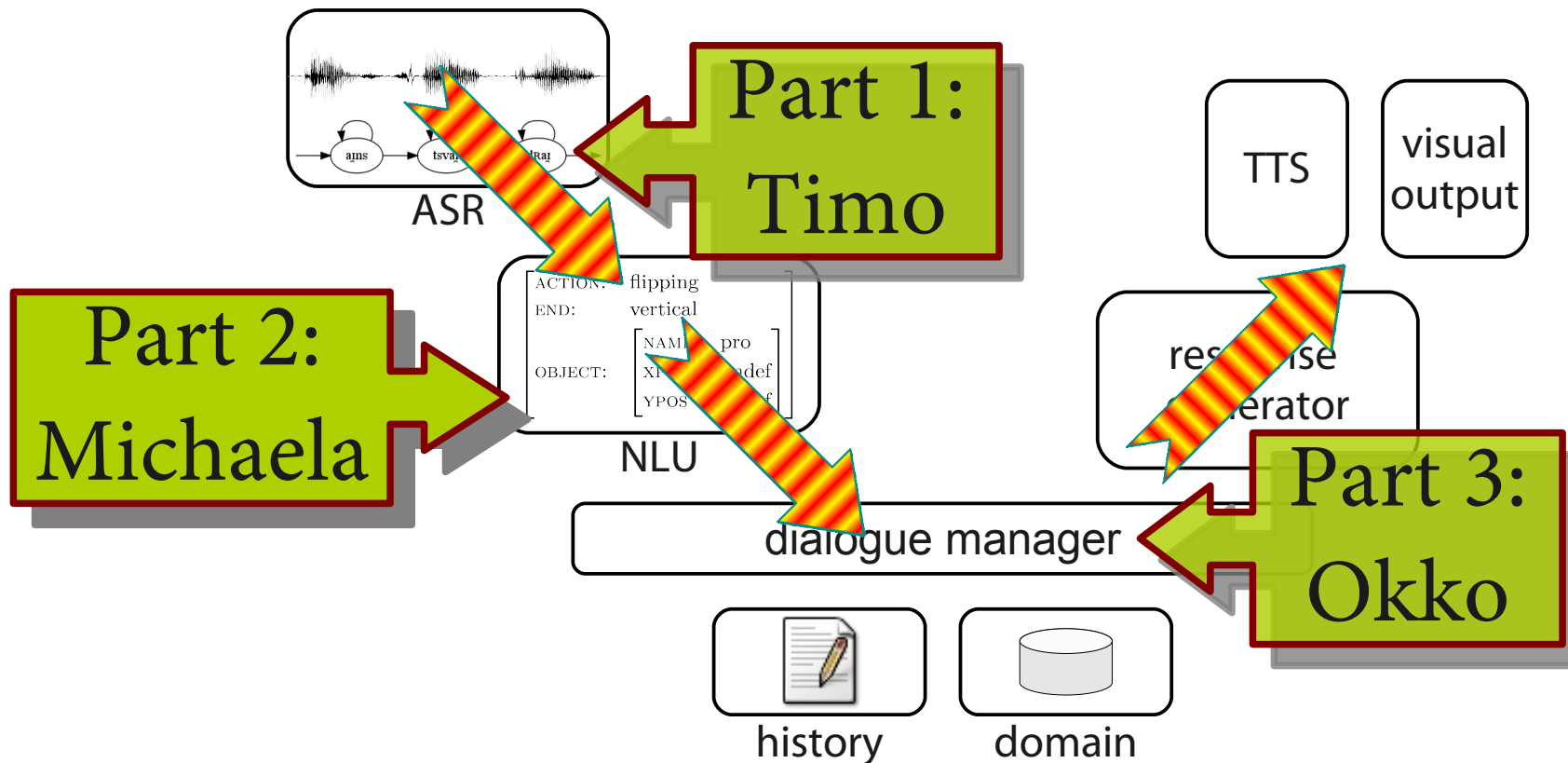
- no reaction before the user finishes talking
-

Context: **Incremental** Spoken Dialogue Systems



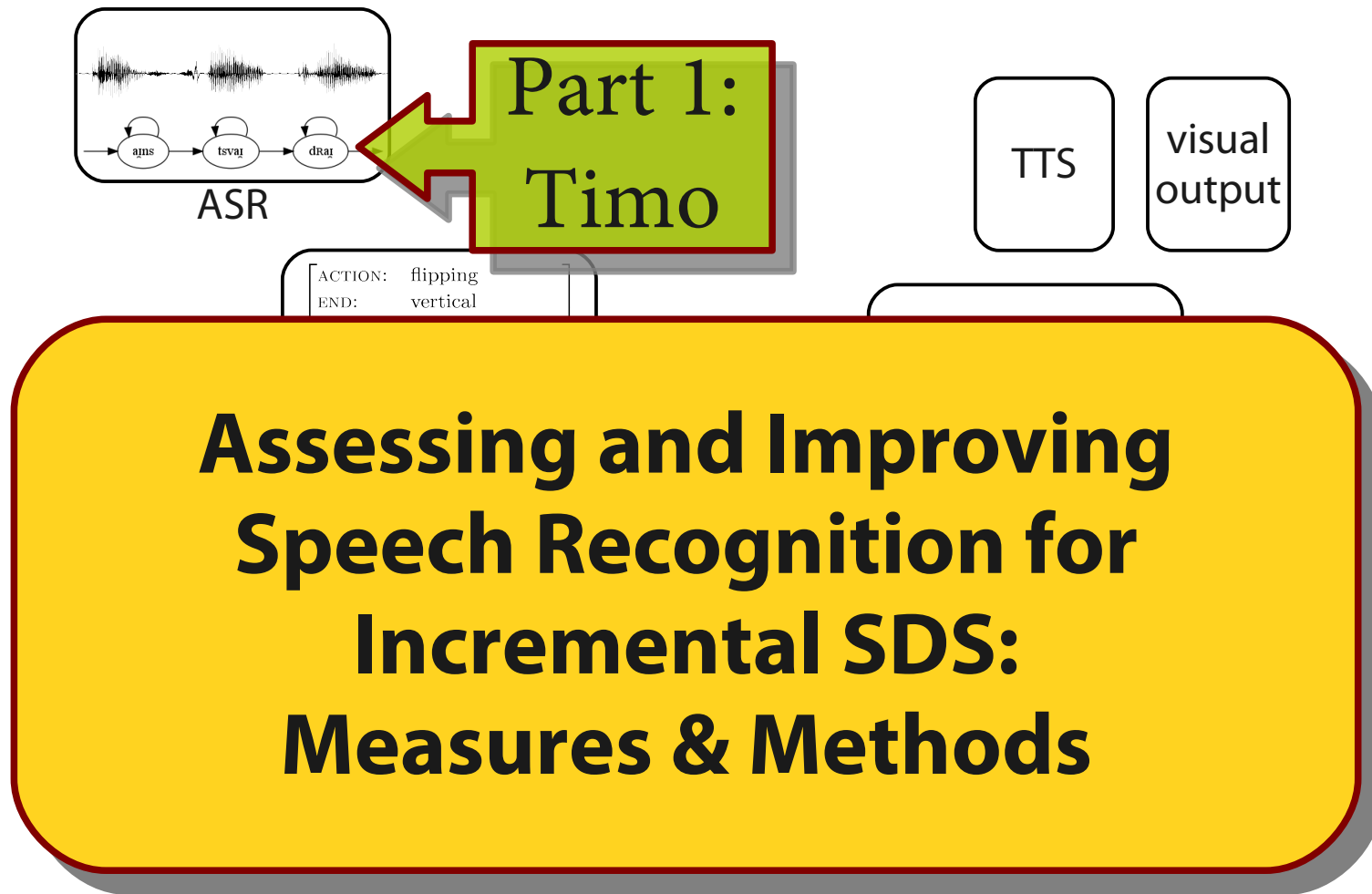
- **partial results** are being processed immediately
- reaction is quicker, back-channels are possible

Context: **Incremental** Spoken Dialogue Systems



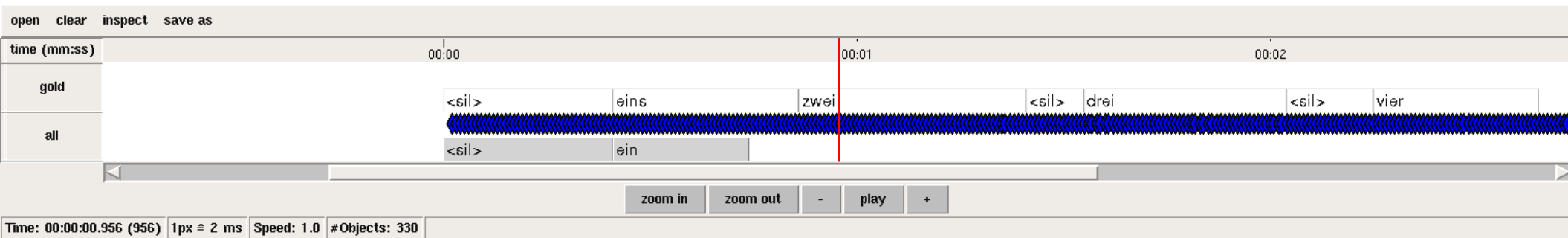
- **partial results** are being processed immediately
- reaction is quicker, back-channels are possible

Context: **Incremental** Spoken Dialogue Systems



A Real-World Example of Incremental ASR Hypotheses

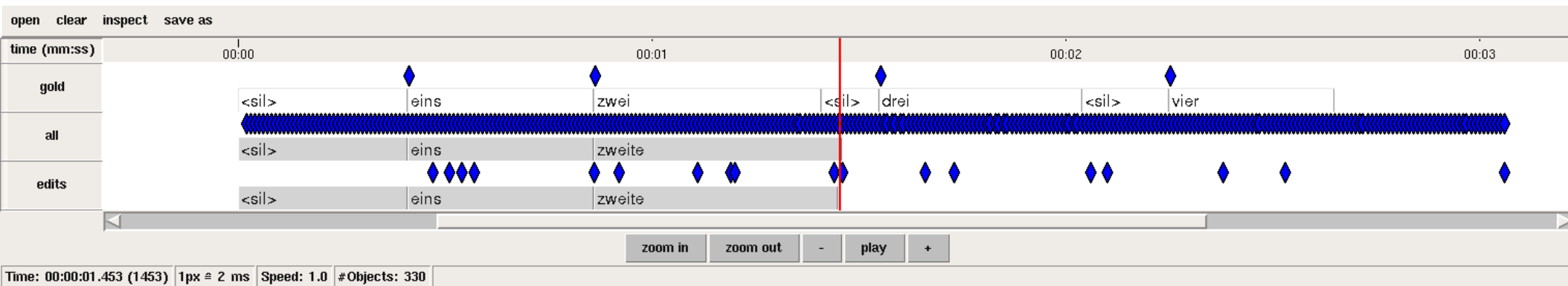
Software from Malsburg et al., submitted



- ASR hypotheses change with time (open movie)

A Real-World Example of Incremental ASR Hypotheses

Software from Malsburg et al., submitted

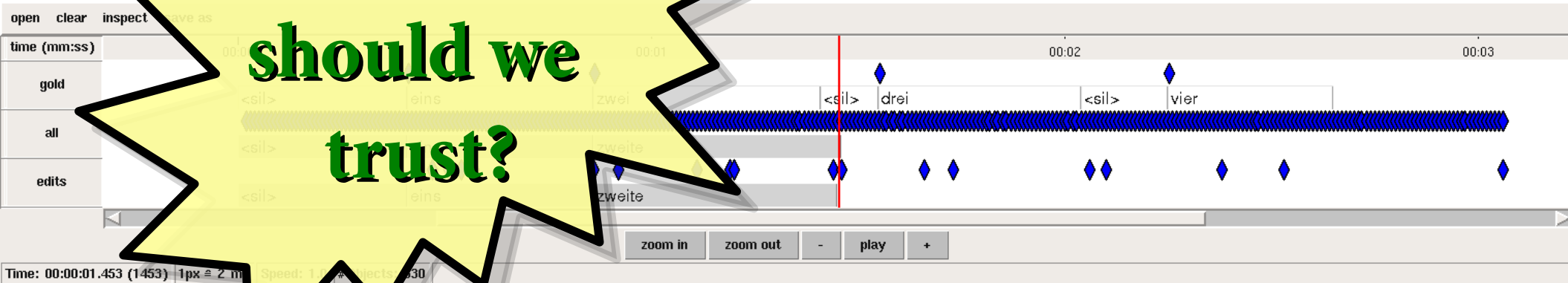


- ASR hypotheses change with time
- more edit than necessary → **overhead ~ 90% !**
 - 90% of a consumers work will be **useless**

A Real-World Example of Incremental ASR Hypotheses

**which edits
should we
trust?**

Software from Malsburg et al., submitted



- ASR hypotheses change with time
- more edit than necessary → overhead ~ 90 % !

A Real-World Example of Incremental ASR Hypotheses

Software from Malsburg et al., submitted

**which edits
should we
trust?**

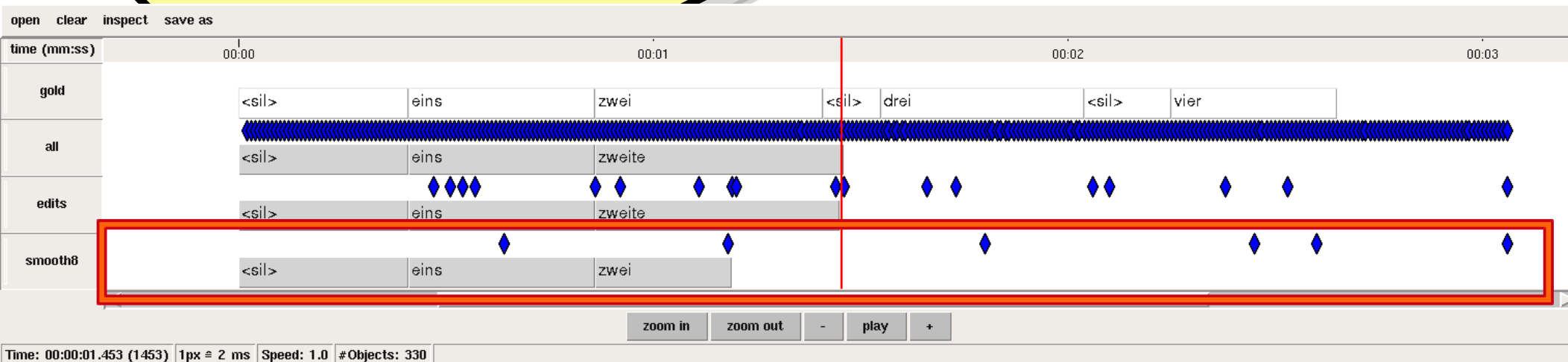
**Patience,
Young Jedi!
waiting helps**

- ASR hypotheses change with time
- more edit than necessary → overhead ~ 90%!
- reduce overhead, sacrifice some timeliness

A Real-World Example of Incremental ASR Hypotheses

which edits

Software from Malsburg et al., submitted



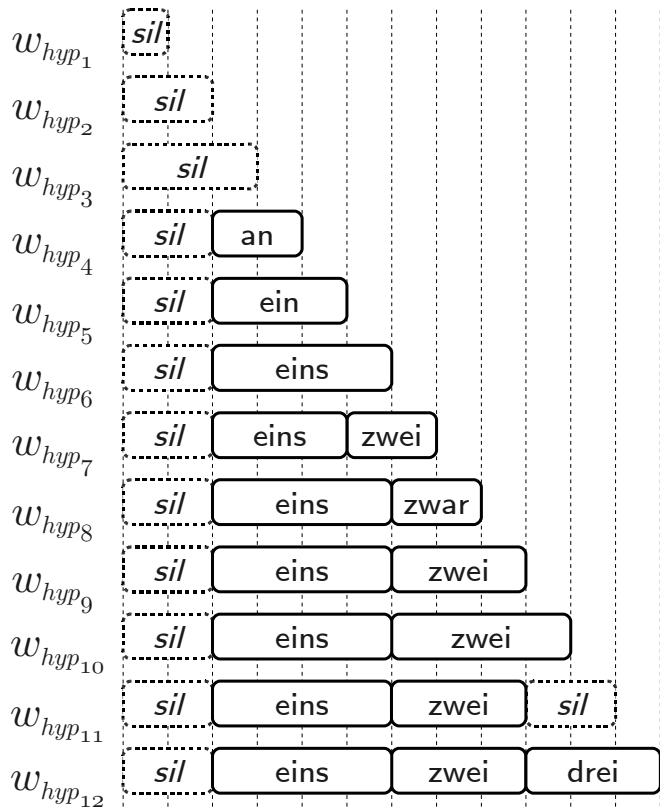
waiting helps

- ASR hypotheses change with time
- more edit than necessary → overhead ~ 90% !
- reduce overhead, sacrifice some timeliness

A Reduced Example

w_{gold} sil eins zwei drei ...

time: 0 1 2 3 4 5 6 7 8 9 10 11 12



$\oplus(\text{an})$

$\ominus(\text{an}), \oplus(\text{ein})$

$\ominus(\text{ein}), \oplus(\text{eins})$

$\oplus(\text{zwei})$

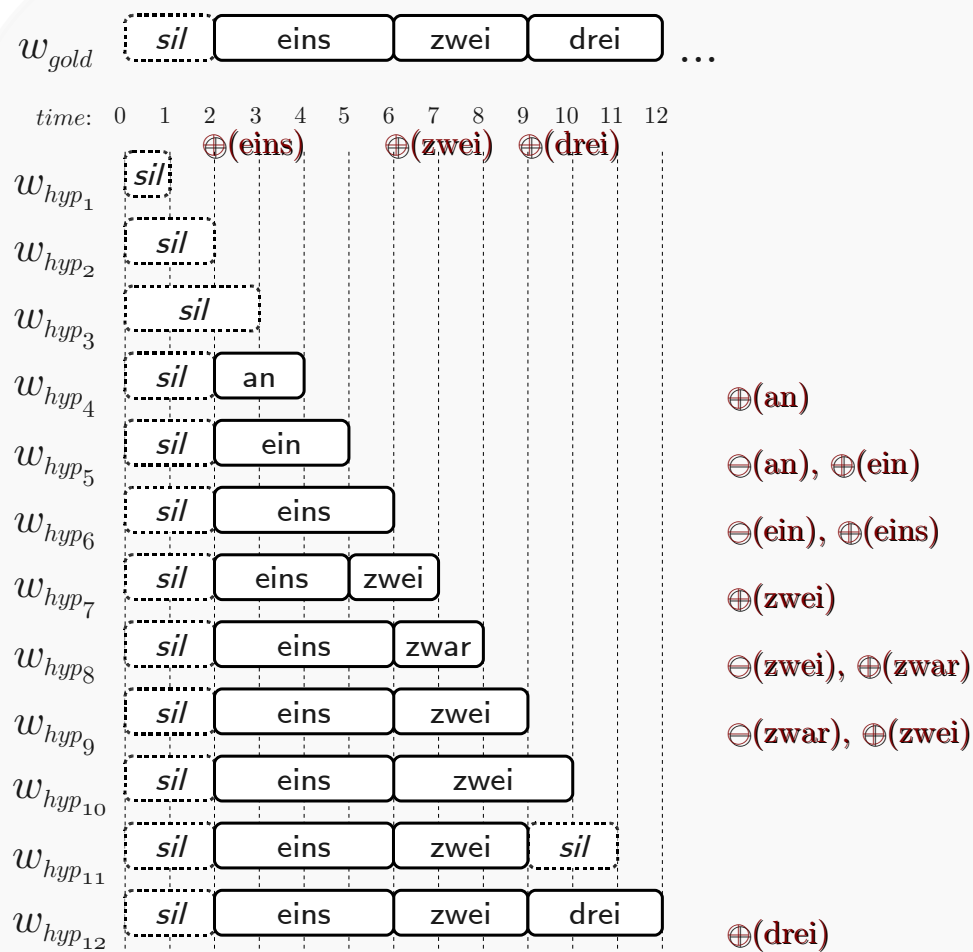
$\ominus(\text{zwei}), \oplus(\text{zwar})$

$\ominus(\text{zwar}), \oplus(\text{zwei})$

$\oplus(\text{drei})$

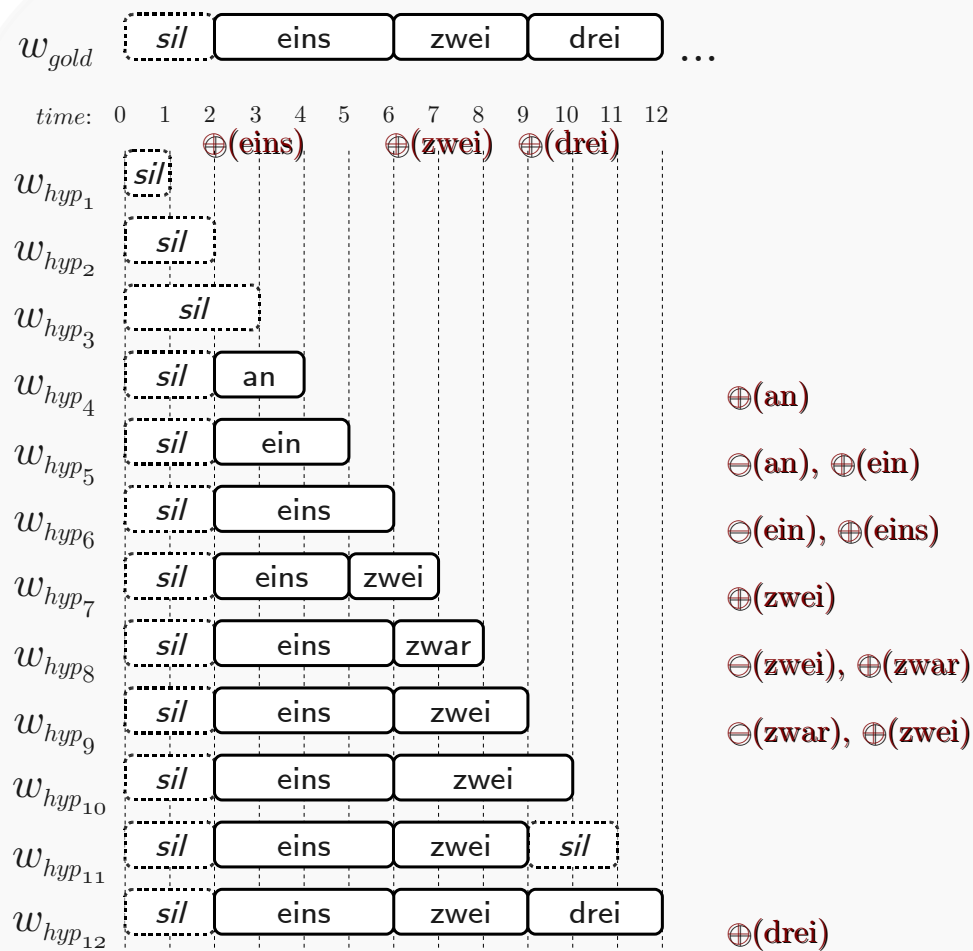
- w_{hyp_t} is the word sequence hypothesized at time t
- two dimensions:
 - time we reason about: \rightarrow
 - time we reason at: \downarrow
- w_{gold} is final hypothesis

Change Measure



- changes on the right
- *add, delete or revise*
- ideally: one *add* per word
- in fact: **edit overhead**
- $EO = \frac{|unnecessary\ edits|}{|edits|}$

Change Measure



ideally: 3 edits

actually: 11 edits

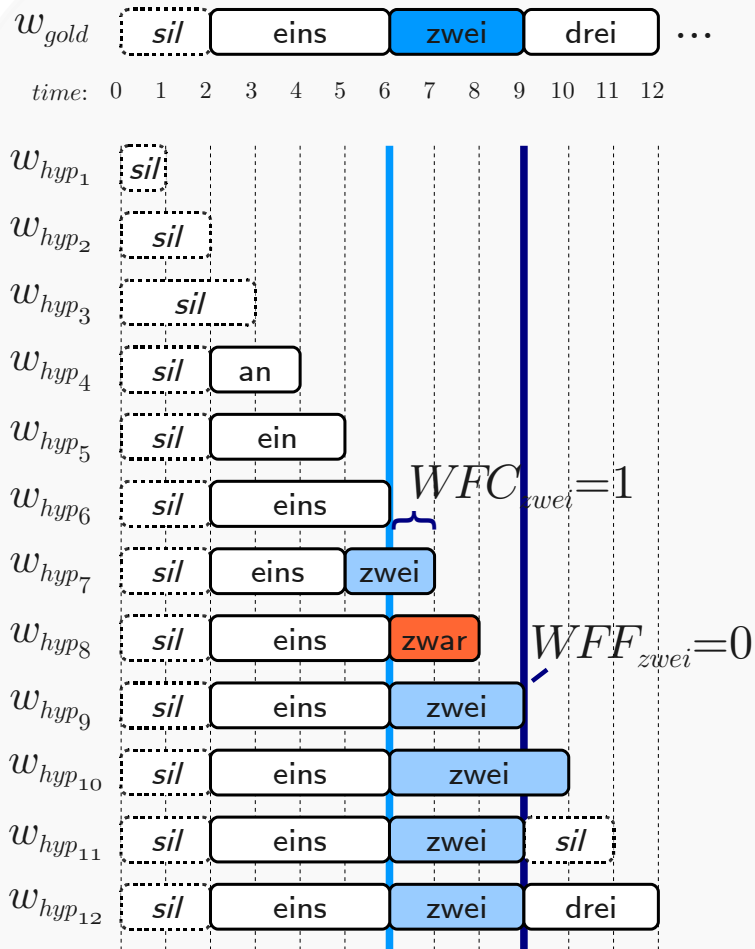
unwanted: 8 edits

EO: $8/11 = 72\%$

Edits are bad:

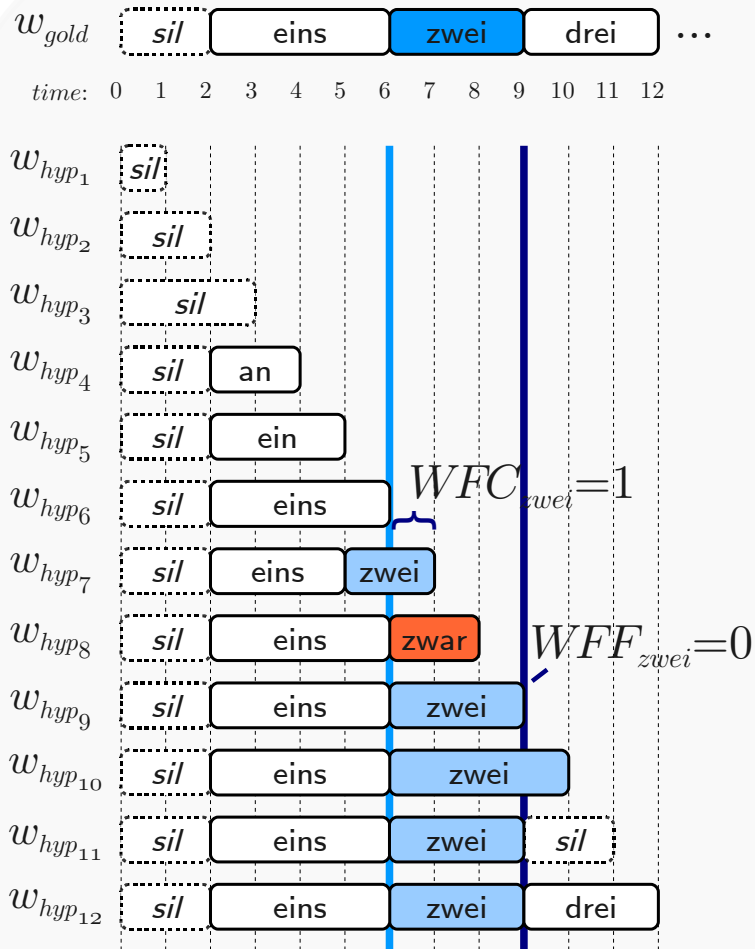
- edits lead to unnecessary processing of a consumer
 - less edits mean less processing
 - we would like to **reduce the edit overhead**
 - by **deferring** or **suppressing** edits
 - deferring edits leads to delays,
deteriorating *timing measures* ...
-

Measuring Timing



- when do we find out about a word?
 - word first correct: **WFC**
- when do we become certain about a word?
 - word first final: **WFF**
- this is per word
 - averages are important

Measuring Timing



- when do we find out about a word?

- word **first** WFC

- when do we become certain about a word?

- word **final** WFF

- this is per

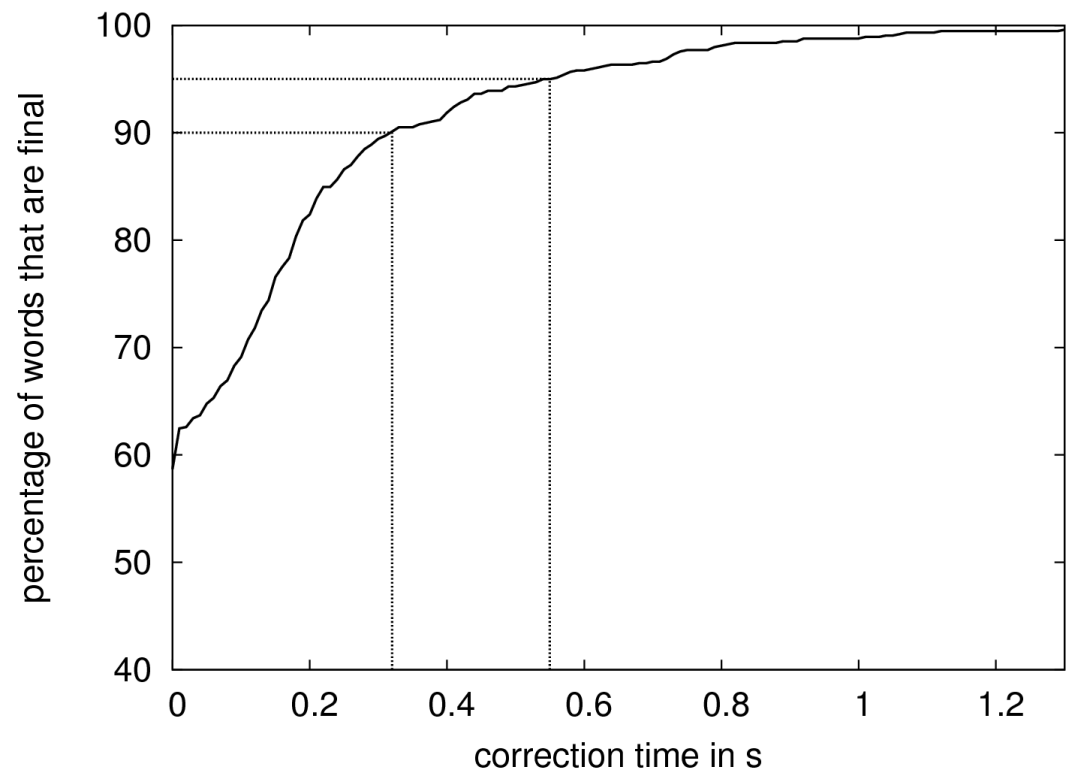
→ averages are important

Timing Measures

- depending on the use-case we may care for ...
 - if we want to **assume** as soon as possible → low **first**
 - if we want to **know** as soon as possible → low **final**
 - deferring edits means two things:
 - worse **first** timings
(as the lag passes through)
 - less increase in **final** timings,
(if we eliminate wrong edits)
-

Certainty Considerations

- the **correction time** for a word is **WFF–WFC**
- 58.6 % of all words are immediately correct
- we can calculate the degree of **certainty** for given hypothesis ages
- e.g. if a correct hyp. lasts for 0.55 s, we can be certain (95 %) that it will not change anymore



Improving Incremental ASR

- our primary goal is to reduce **edit overhead**
 - ... by deferring or suppressing edits
 - deferring edits will always hurt **first** timings
 - less impact on **final** timings
 - the final (non-incremental) result does not change
 - only **trust older parts** of hyps. (Right Context)
 - only **trust older edits** (Message Smoothing)
-

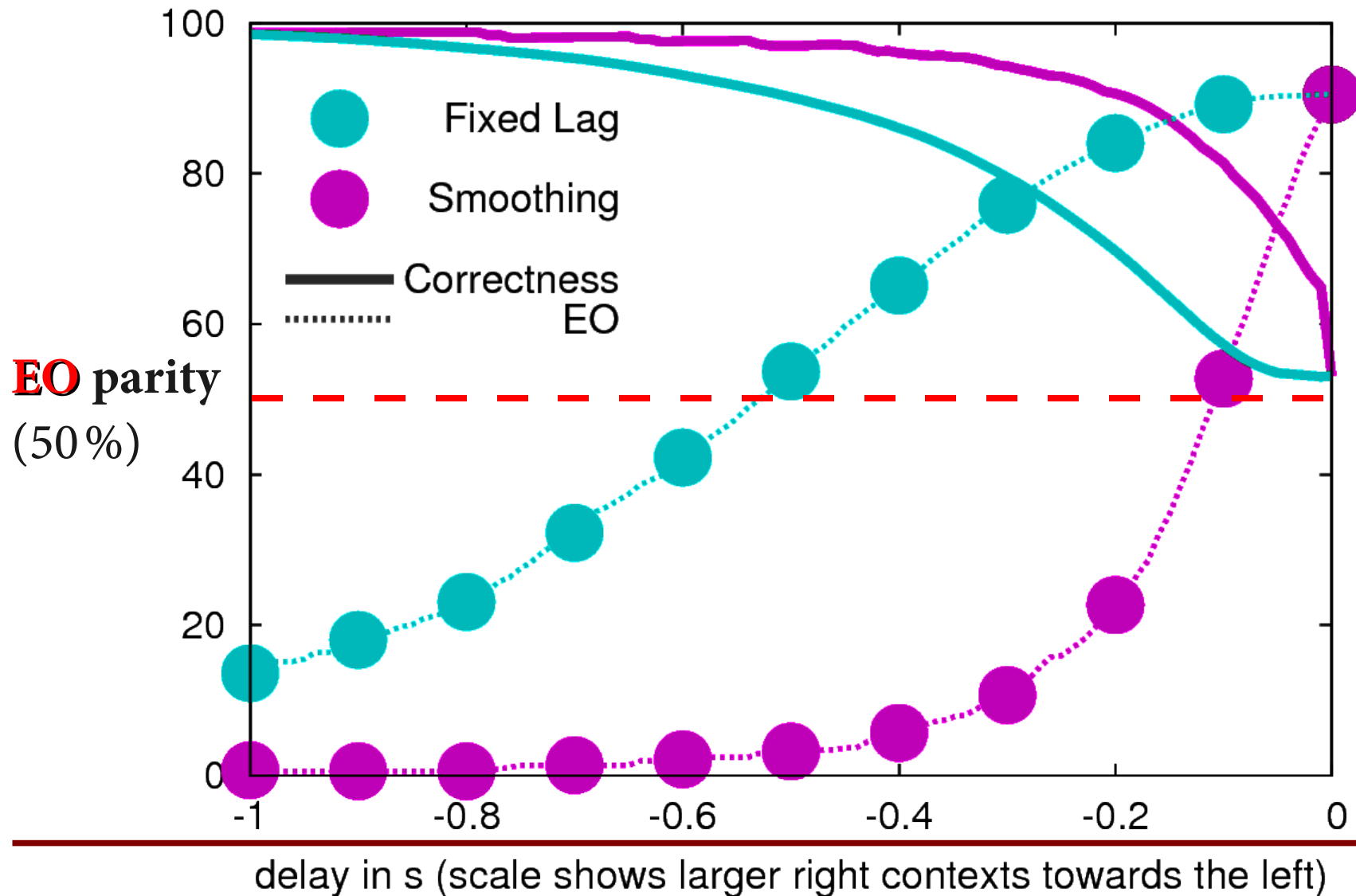
Right Context to Improve Incremental Performance

- much jitter is at the right end of the hypotheses
 - at time t only evaluate hyp_t up to $t-\Delta$
 - we need to take this into account for correctness:
 - *fair* r-correct: $w_{hyp_{t-\Delta}} = w_{gold_{t-\Delta}}$
 - **first** increases with Δ , **final** increases $\leq \Delta$
 - we can **predict the future** with negative Δ
 - e.g. fair r-correctness down 50% at 100ms in the future
-

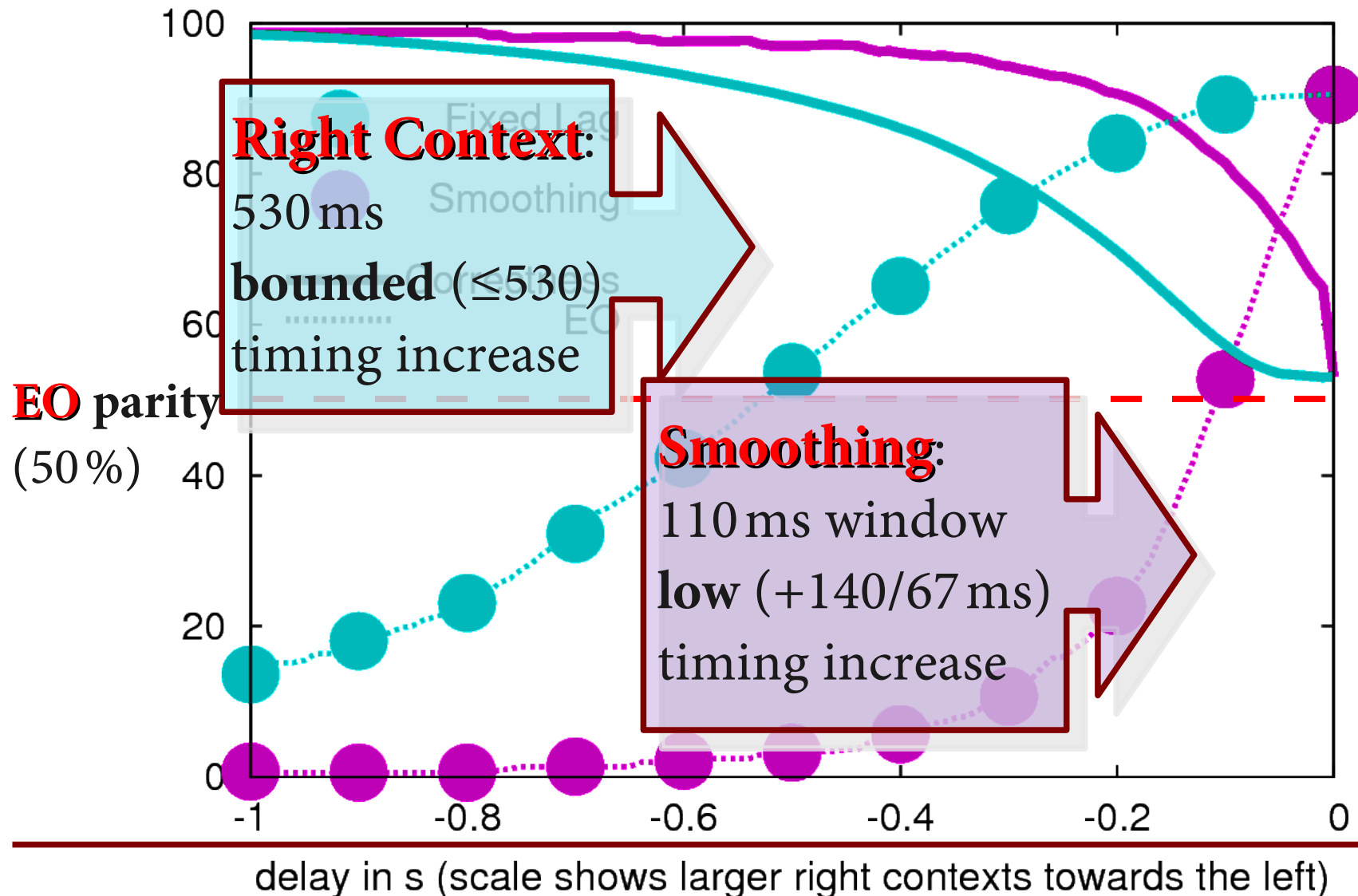
Message Smoothing to Improve Incremental Performance

- most bad edits only last for a short while
 - "zwei" → "zwar" → "zwei"
 - hold back edits until they reach a certain age
 - only output if they don't die before maturing
 - multiple short edits of a word may delay messages:
 - **first** may grow without fixed bounds occasionally
 - probable resolution/mitigation: **future work**
allow for some kind of "majority smoothing"
-

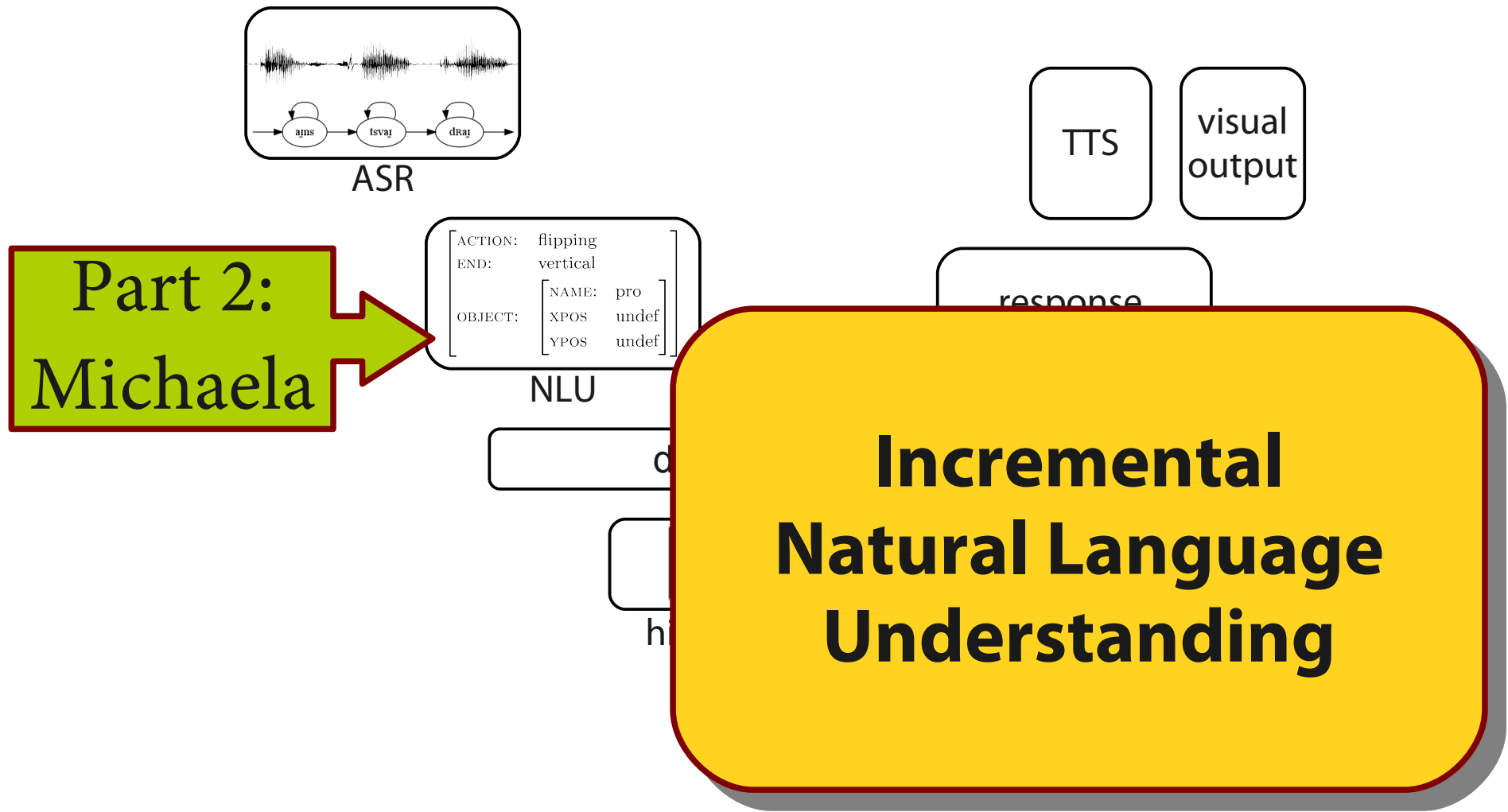
Right Context vs. Smoothing



Right Context vs. Smoothing



Context: **Incremental** Spoken Dialogue Systems



Incremental NLU in INPRO P2

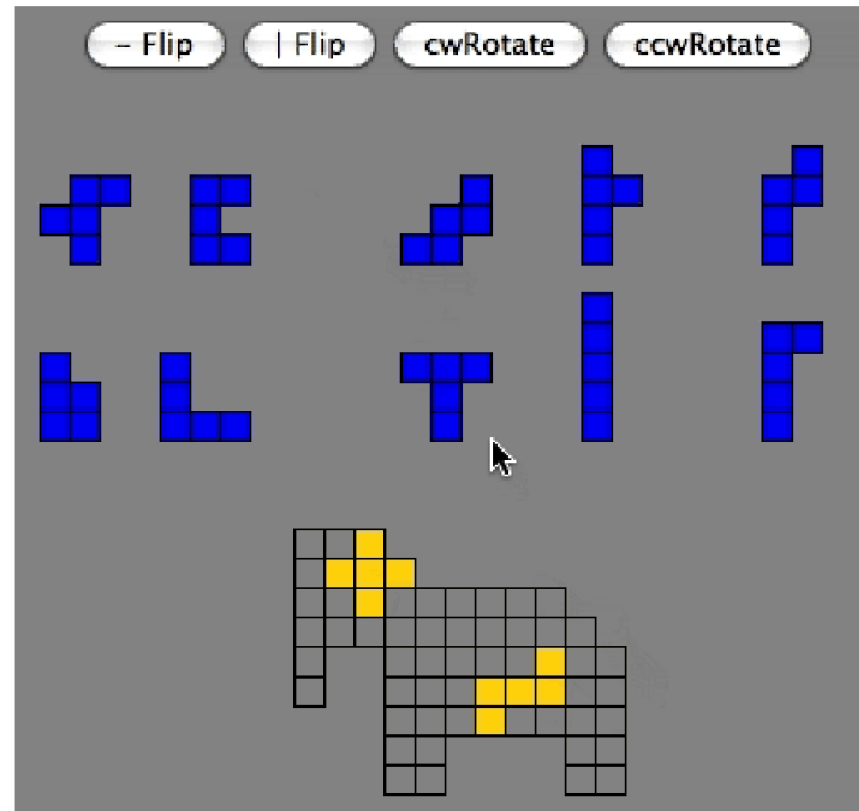
- RUBISC: Robust Unification-Based Incremental Semantic Chunker (SRS� 2009)
 - Incremental Probabilistic Reference Resolver (submitted)
-

RUBISC: Incremental Chunking with Semantic Chunks

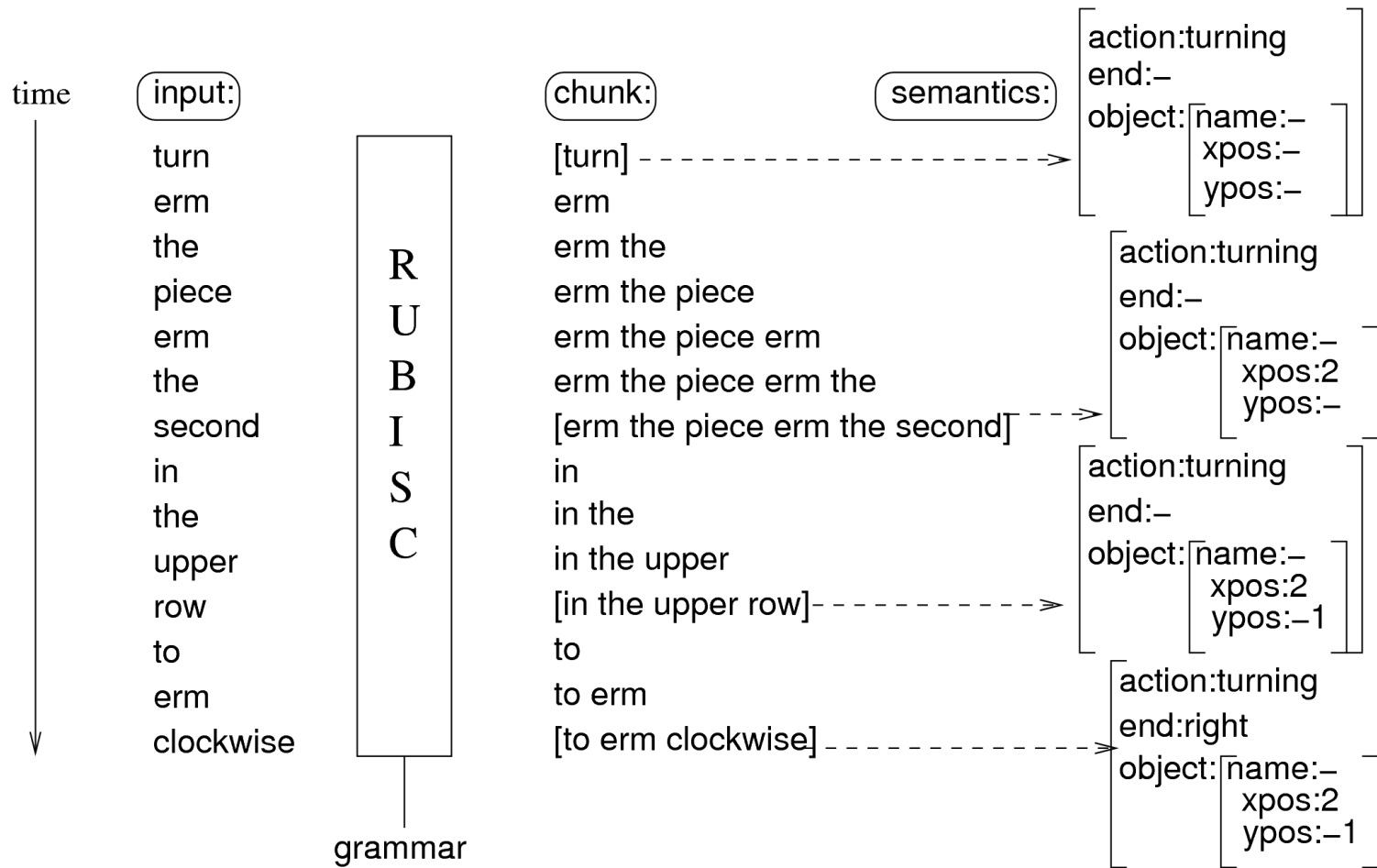
- Chunks based on semantic content rather than syntax
 - Inspired by the notion of sense units (Selkirk, 1984)
 - φ -phrases: consist of head and all its specifiers/head and all the material on the non-recursive side of the head up to the next head outside of its maximal projection (Nespor and Vogel, 1986)
 - Sense units: up to head; semantic chunks: up to semantically relevant material
-

Domain (revisited)

- Actions:
grasp, turn, flip, move
- Objects:
w, cross, ...
- End positions
head, leg, ...



Incremental Chunking



Grammar

@:action

@:entity:name

@:entity:xpos

@:entity:ypos

@:end

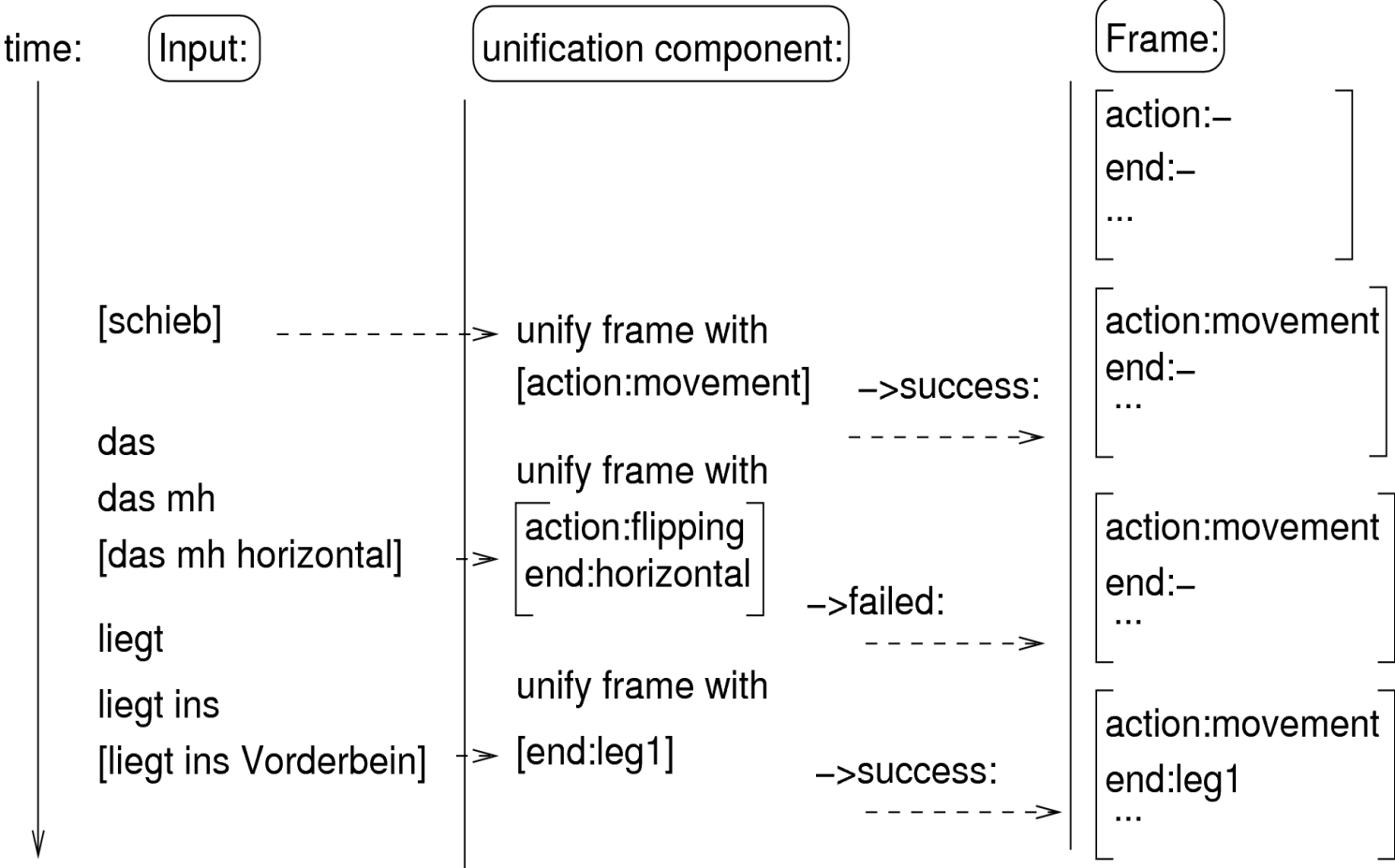
action:flipping-> spieg(le|el) *{flip}*

action:grasping,end:empty -> nimm|nehme *{take}*

entity:name:x -> kreuz|plus|((das|ein) x) *{cross|plus|((the|an) x)}*

end:horizontal,action:flipping -> horizontal *{horizontally}*

Slot Unification



Making Turn-Taking Decisions

Schieb ⚡ *das Kreuz* ⚡ *in den Kopf* ⚡ *des Tieres, das wie ein Elefant ...*

Push the cross into the head

of the animal, which looks like....

Atterer, Baumann, Schlangen
(Coling 2008): Syntactic features :
don't know if sentence has ended

Atterer & Schlangen (SRSL 2009):
Chunker state: 3 slots filled:
action:movement
object:cross
end:head
-> can react, barge in etc

Implementation

- OAA-agent
 - Supports: add, revoke, commit
 - Evaluation: 55% frames correct, 87% slots correct
-

Incremental Reference Resolver

- Complementary agent to chunker
- Bayesian believe update model which treats the intended referent as a latent variable generating a sequence of observations ($w_{1:n}$ is the sequence of words w_1, w_2, \dots, w_n):

- $$P(r|w_1, \dots, w_n) = \alpha * P(w_n | r, w_1, \dots, w_{n-1}) * P(r|w_1, \dots, w_{n-1})$$

Normalizing factor

Likelihood of the new observation (approximated)

Prior at step n: posterior of previous step

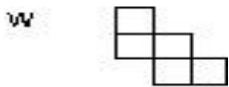
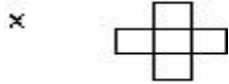
Reference Resolution -- Disfluencies

- Psycholinguistic evidence: more hesitations when describing something hard (Tanenhaus et al., 1995; Brennan & Schober, 2001; Bailey & Ferreira, 2007; Arnold et al., 2007)
 - Include (filled) pauses into our language model
-

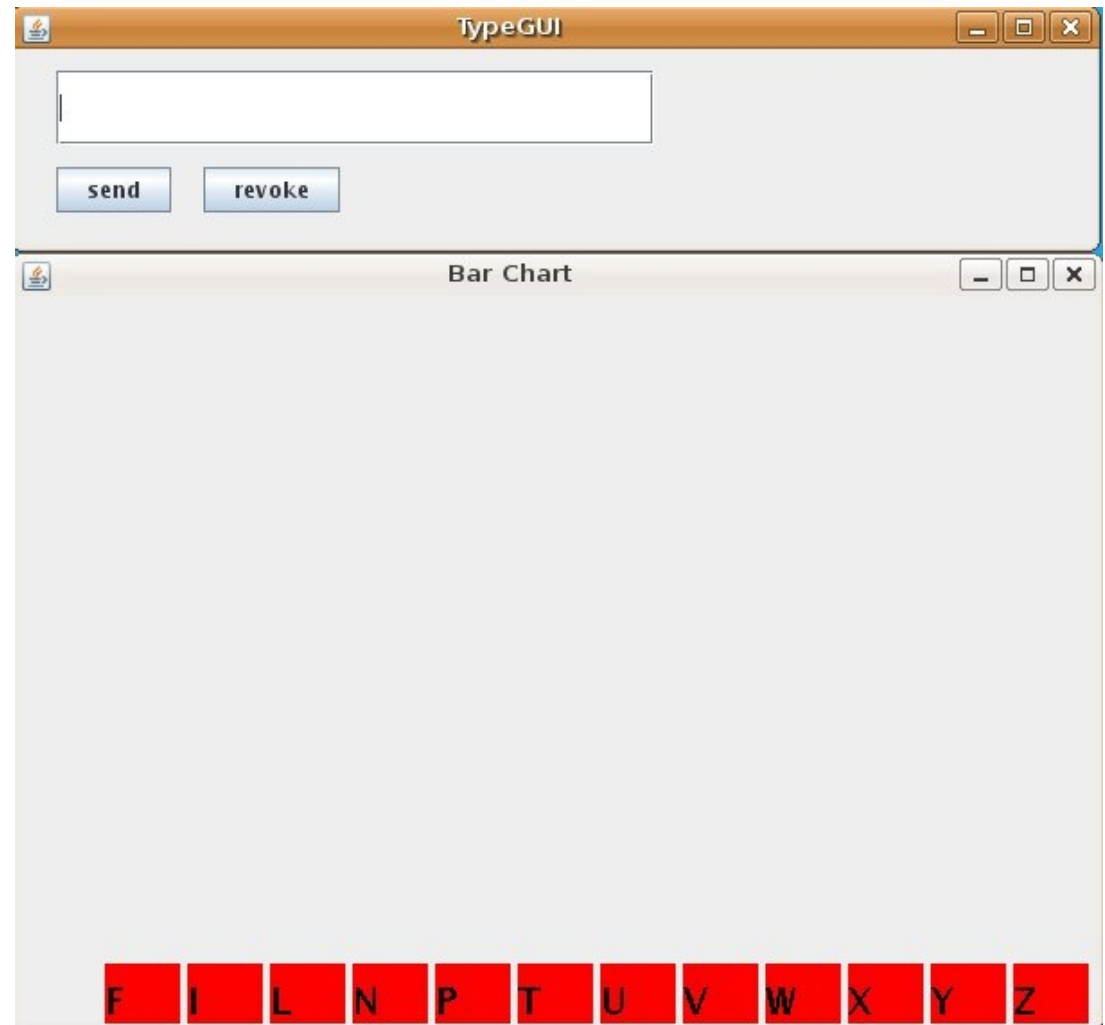
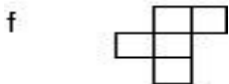
Probabilistic Reference Resolution

- OAA agent demo:

simple:



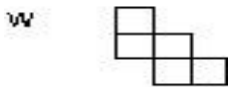
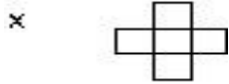
hard:



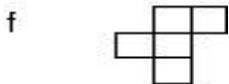
Probabilistic Reference Resolution

- Add 'nimm':

simple:



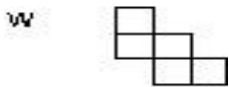
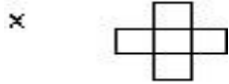
hard:



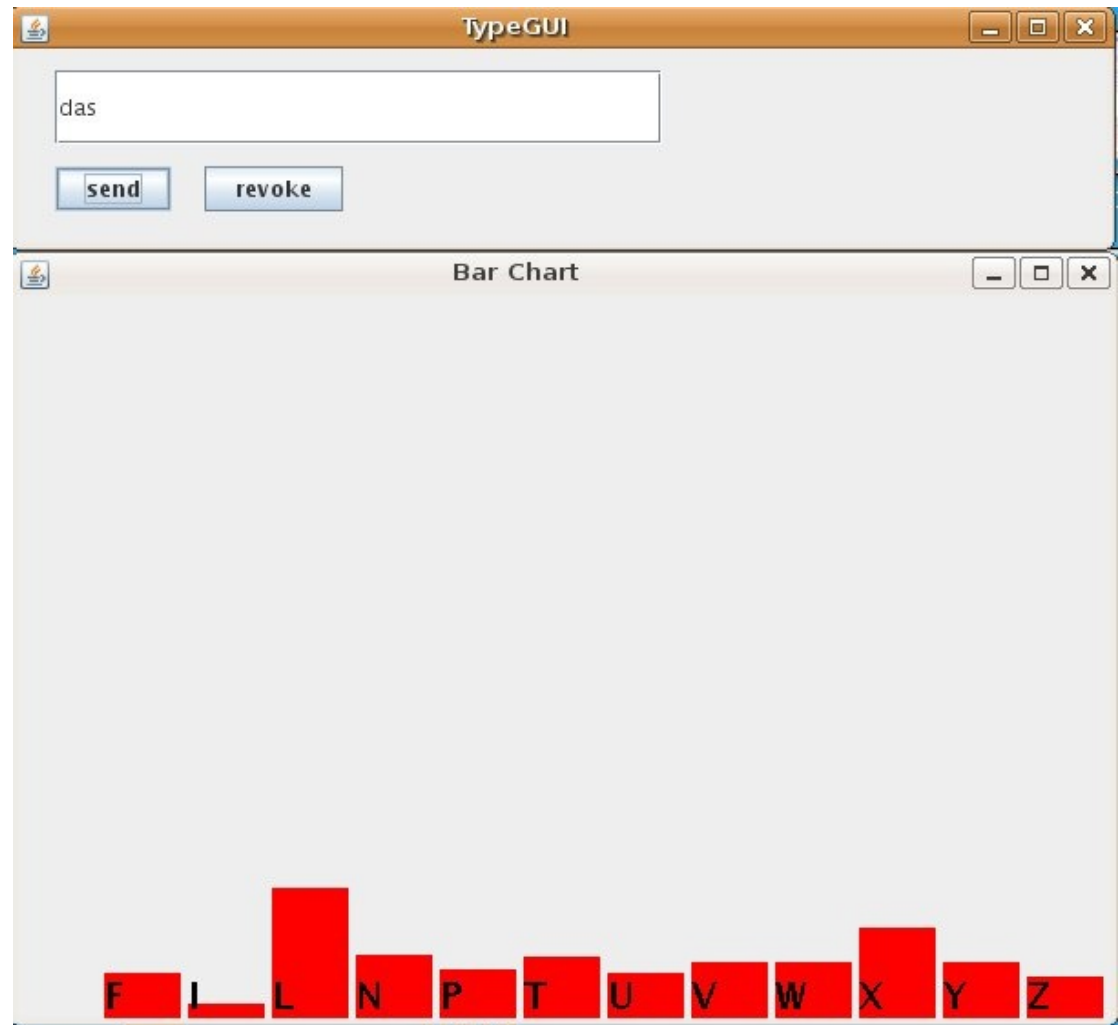
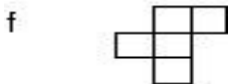
Probabilistic Reference Resolution

- Add 'das':

simple:



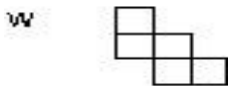
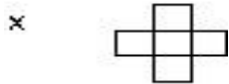
hard:



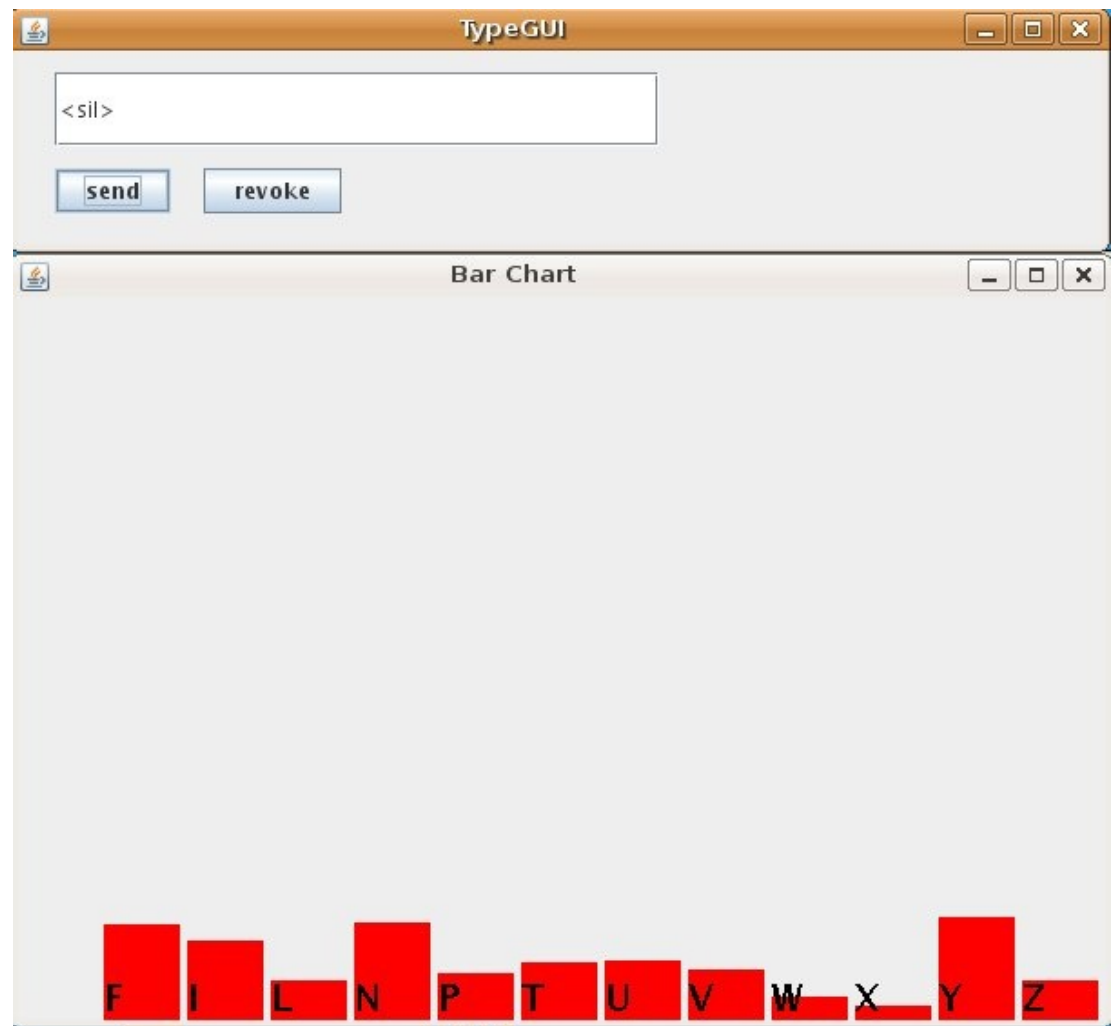
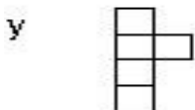
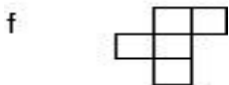
Probabilistic Reference Resolution

- Add '<sil>':

simple:



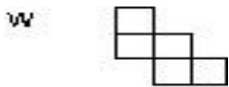
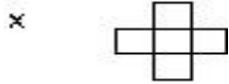
hard:



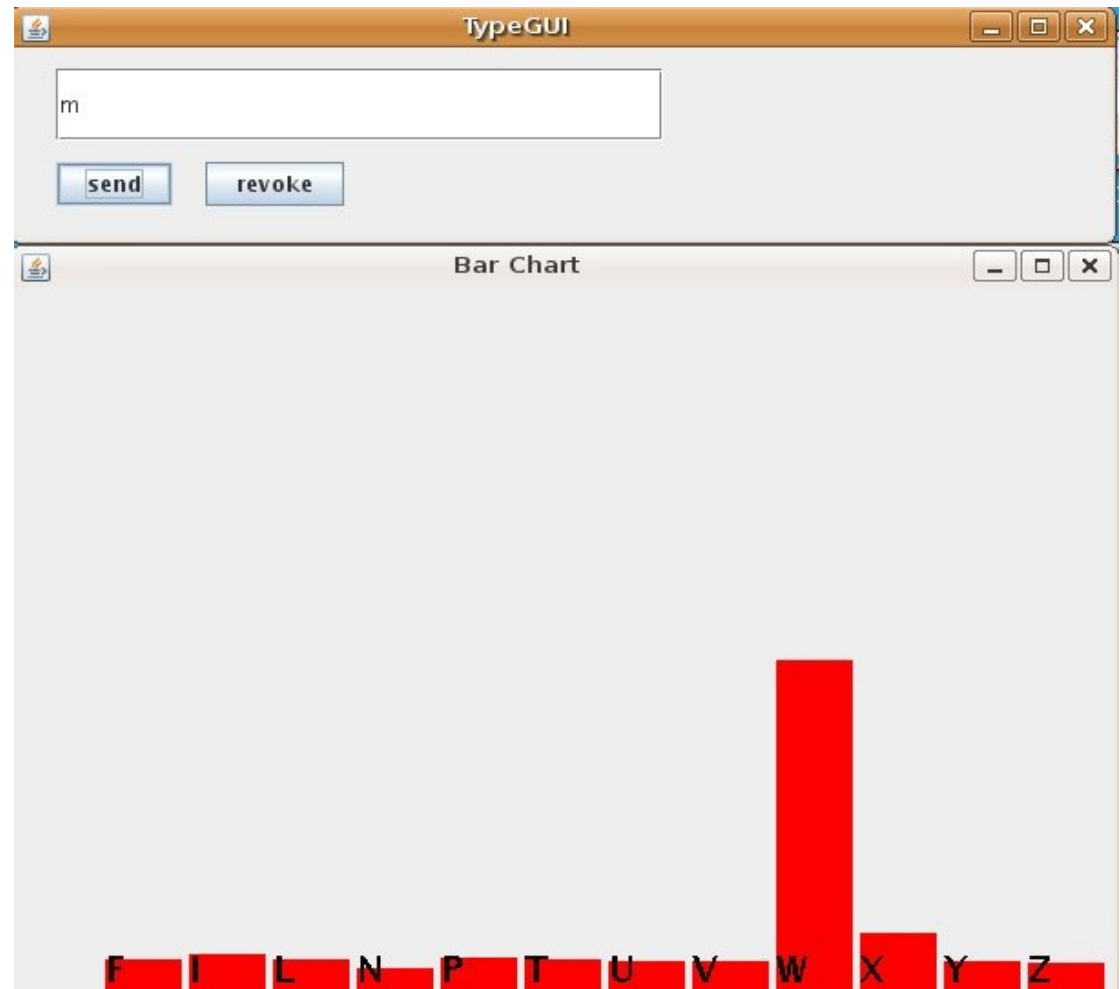
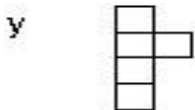
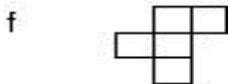
Probabilistic Reference Resolution

- Add 'm':

simple:



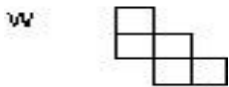
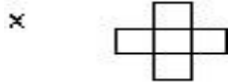
hard:



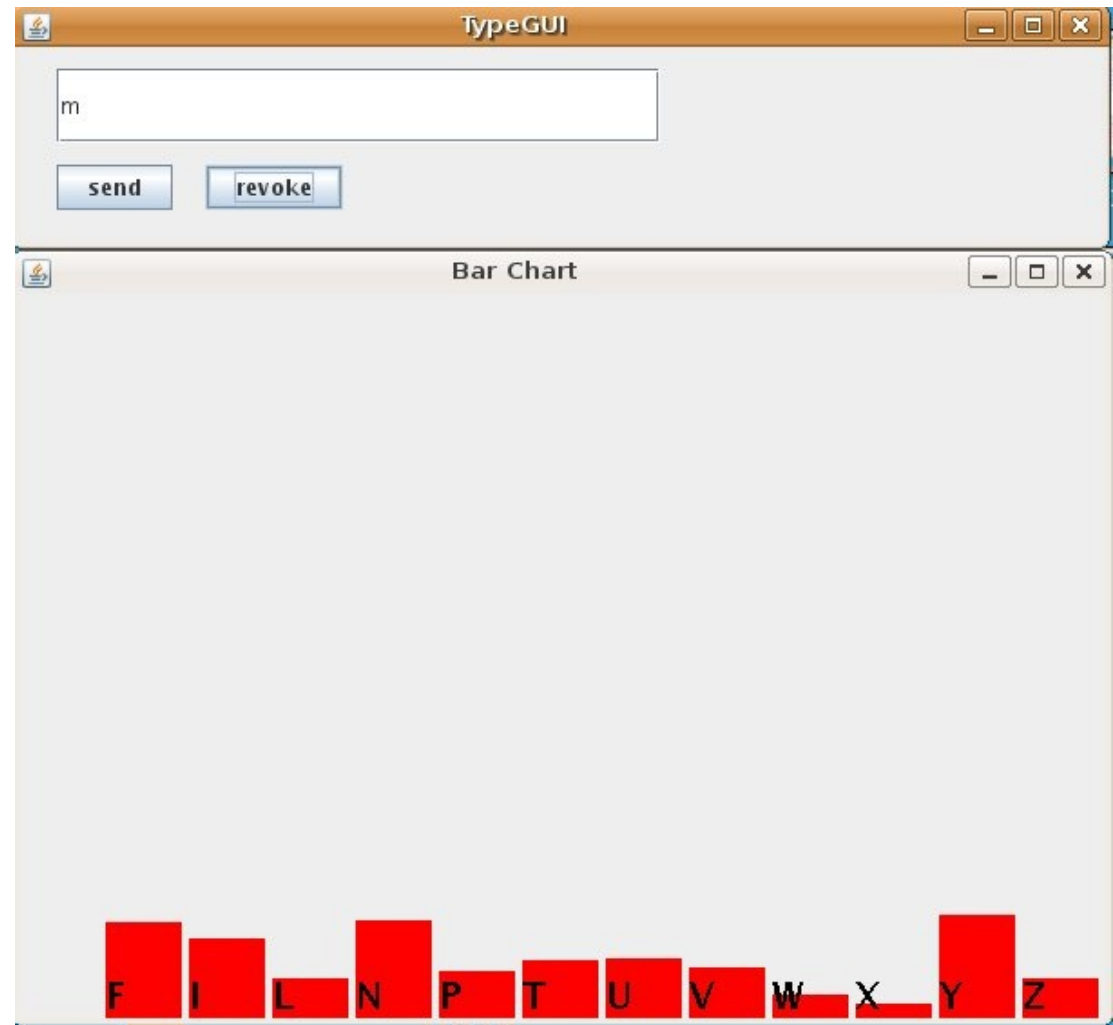
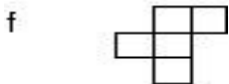
Probabilistic Reference Resolution

- Revoke 'm':

simple:



hard:



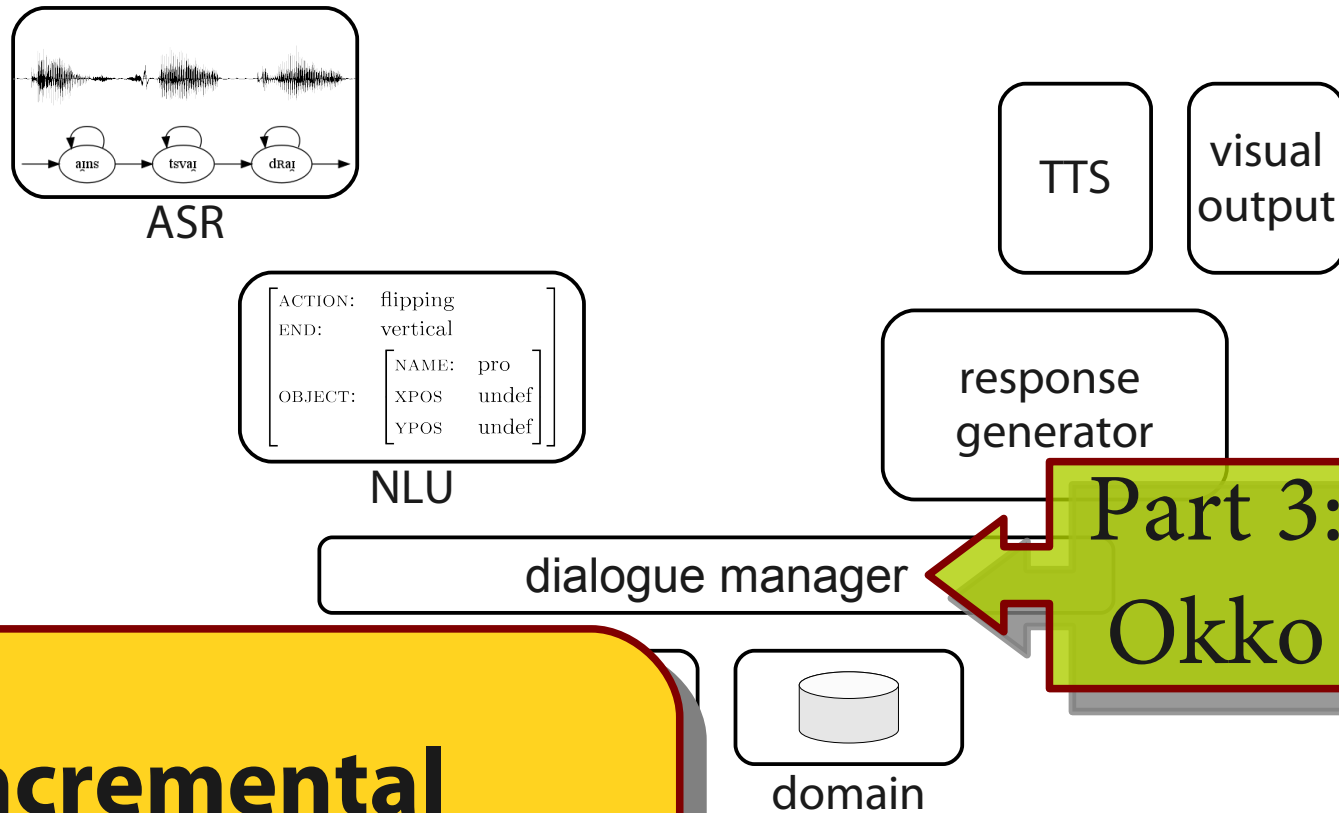
Reference Resolution - Evaluation

- N-best approach: select all pieces above a threshold

	hes	no hes
Last belief correct:	55%	54%
When correct?	88%	85%
correct during silences:	37%	31%

(Schlangen, Atterer, Baumann (submitted))

Context: **Incremental** Spoken Dialogue Systems



Part 3:
Okko

**Incremental
Dialogue
Management**

Dialogue Manager

Overview

- Receives incremental add/revoke/commit messages from acoustic, ASR and NLU components and allows in-utterance processing.
 - Handles timeouts for different kinds of dialogue acts to define behaviour (domain-independent interaction management/dialog skills).
 - Delegates actions across modalities via Action Manager
-

Dialogue Manager

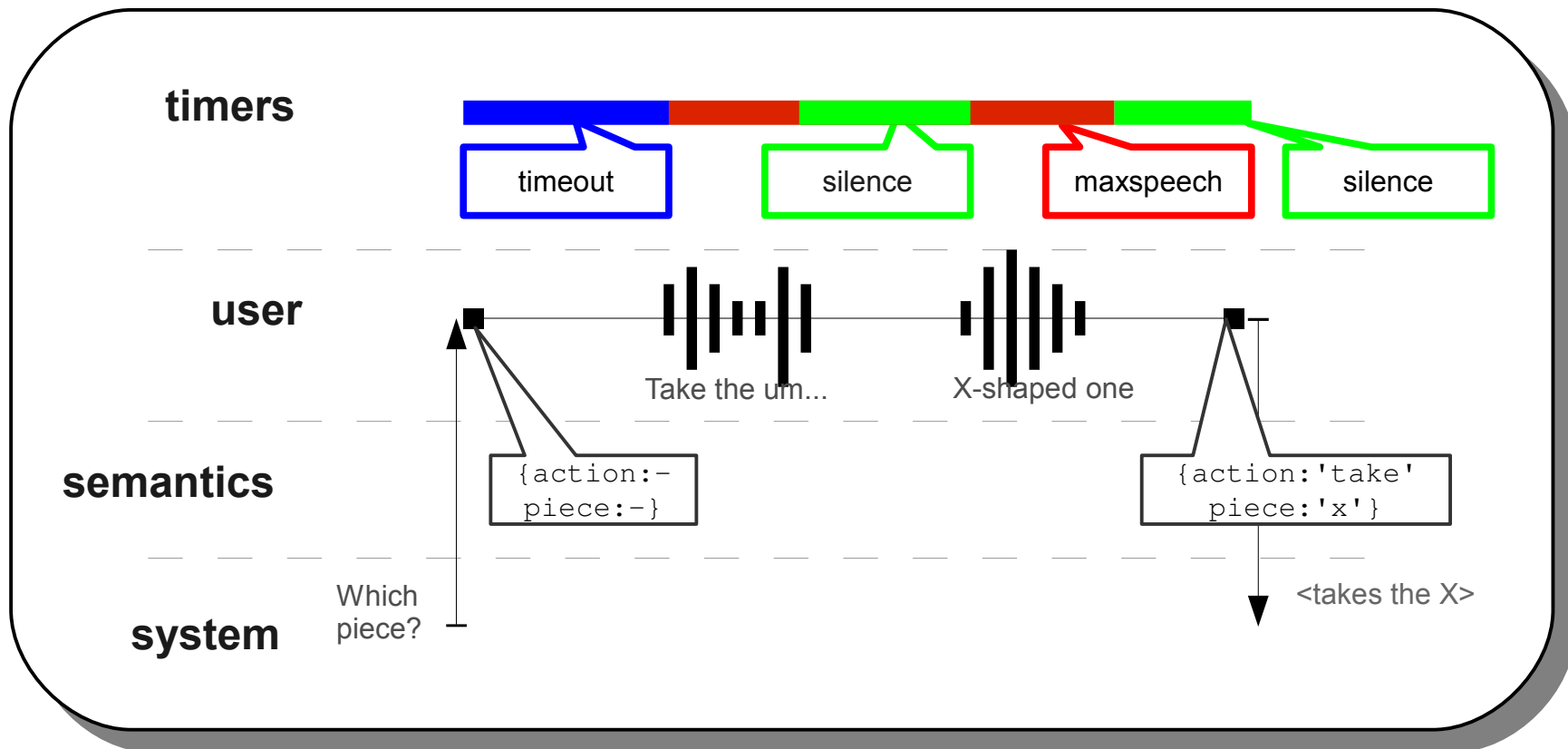
Incremental vs transaction-based DM (1)

- Incremental
 - Events are handled asynchronously.
 - All components are always active, no blocking control.
 - Dialogue state and timeouts must vary dynamically.
 - Transaction-based
 - Events are handled sequentially.
 - Single component has control.
 - Static property set.
-

Dialogue Manager

Incremental vs transaction-based DM (2)

Transaction-based dialog timeouts



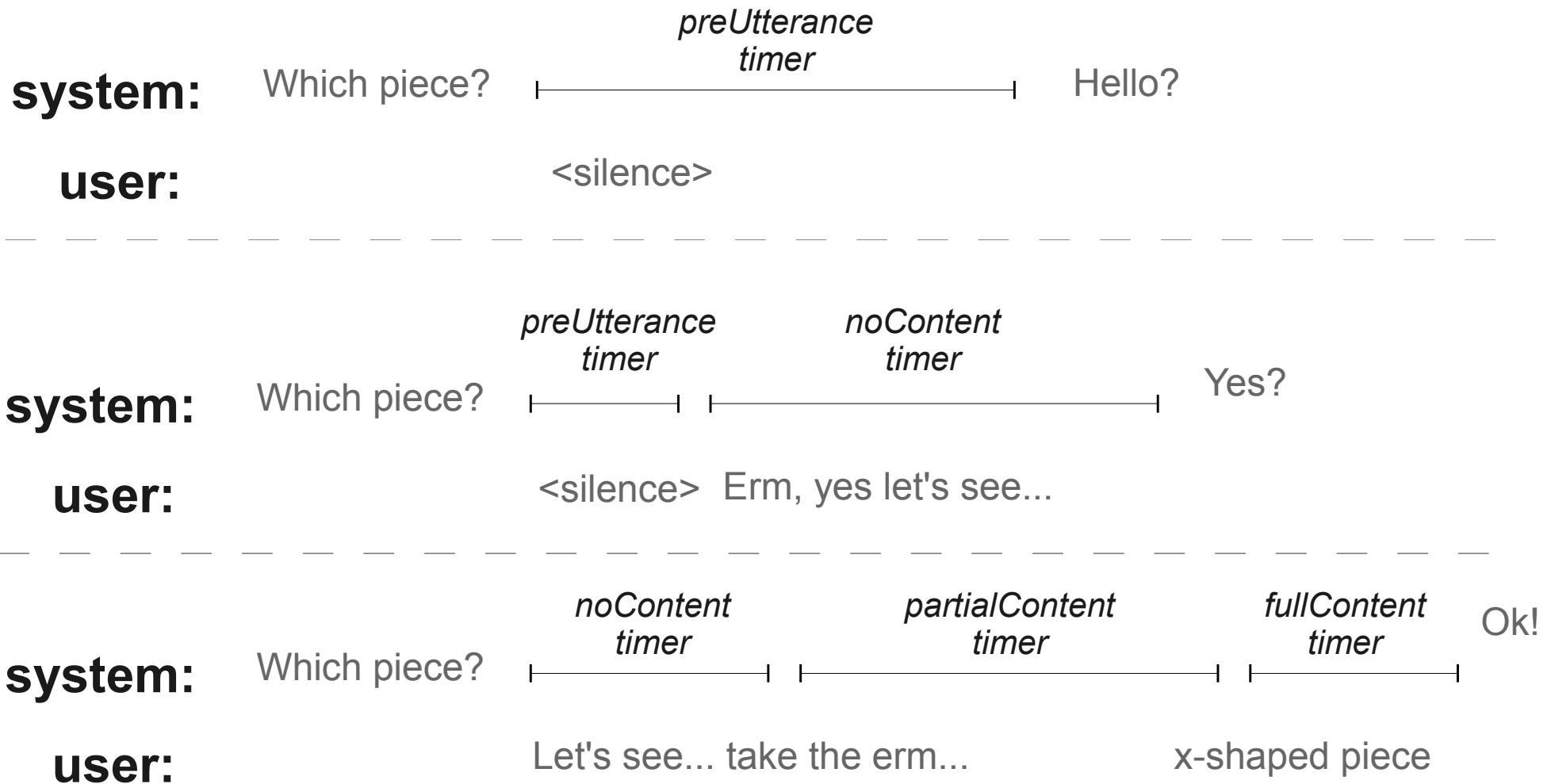
Dialog Manager

Incremental Timers & States (1)

- Timers
 - maxspeech
 - ◆ Cuts off talk that leads nowhere
 - silence
 - ◆ Has settings for types of silence correlating to dialog states
 - States
 - preUtterance
 - noContent
 - partialContent
 - fullContent
-

Dialog Manager

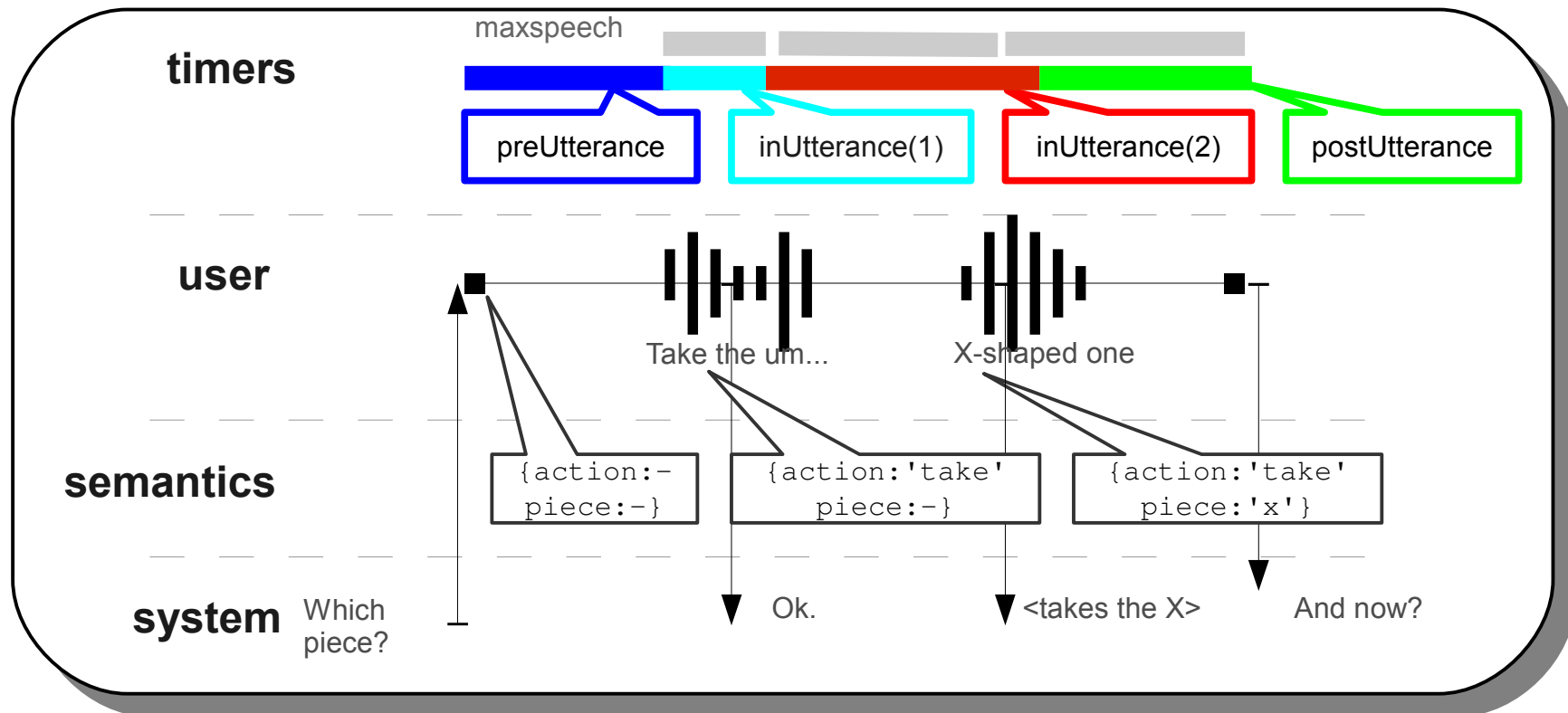
Incremental Timers & States (2)



Dialogue Manager

Incremental vs transaction-based DM (3)

Incremental dialog timeouts



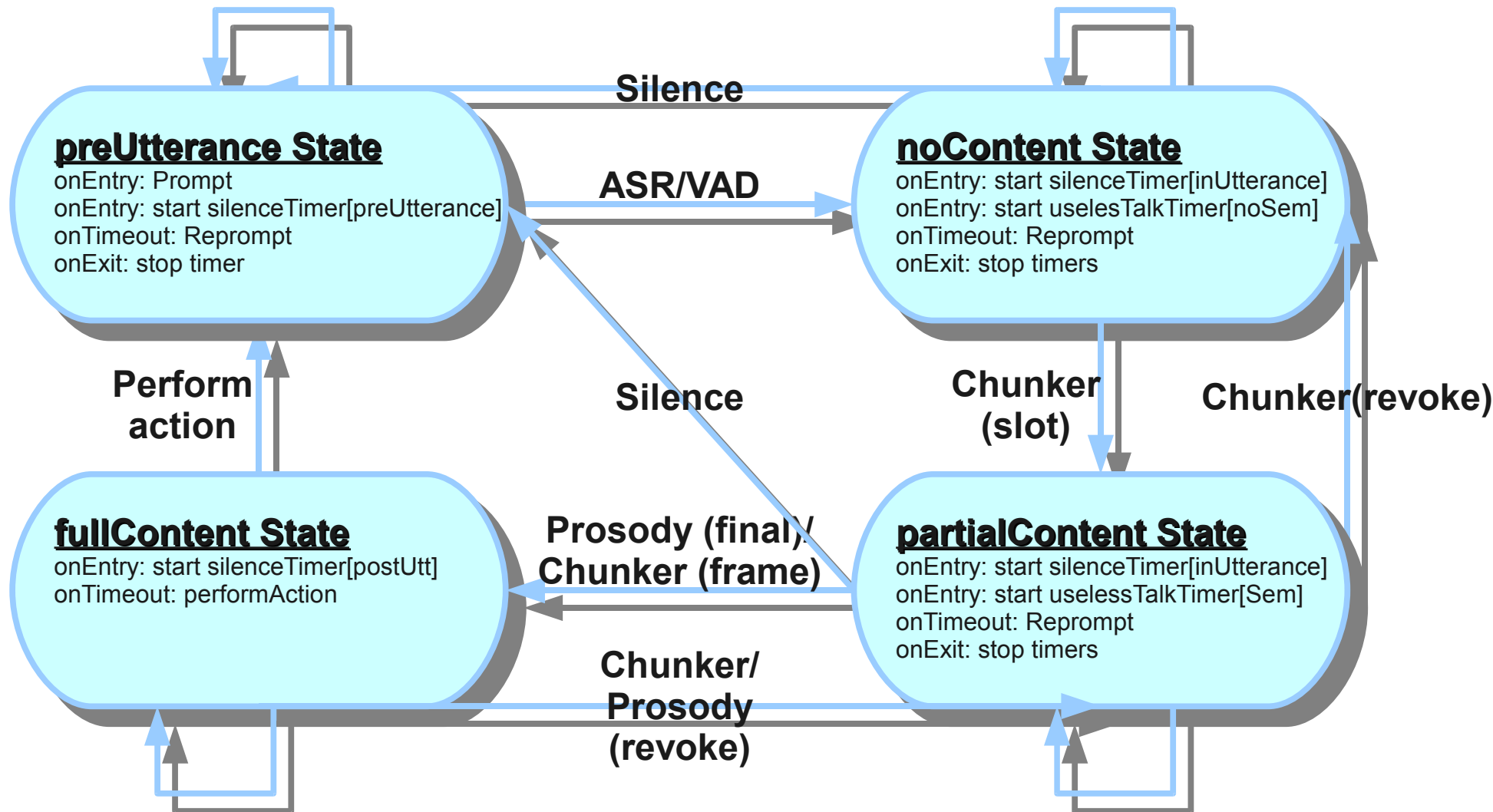
Dialogue Manager

Incremental vs transaction-based DM (4)

- Incremental dialog management has the ability to:
 - process user input before end of utterance.
 - reset timers and thresholds during utterance.
 - issue clarification requests mid-utterance.
 - enable backchannel utterances/dialog acts.
-

Dialogue Manager

Incremental DM State Chart



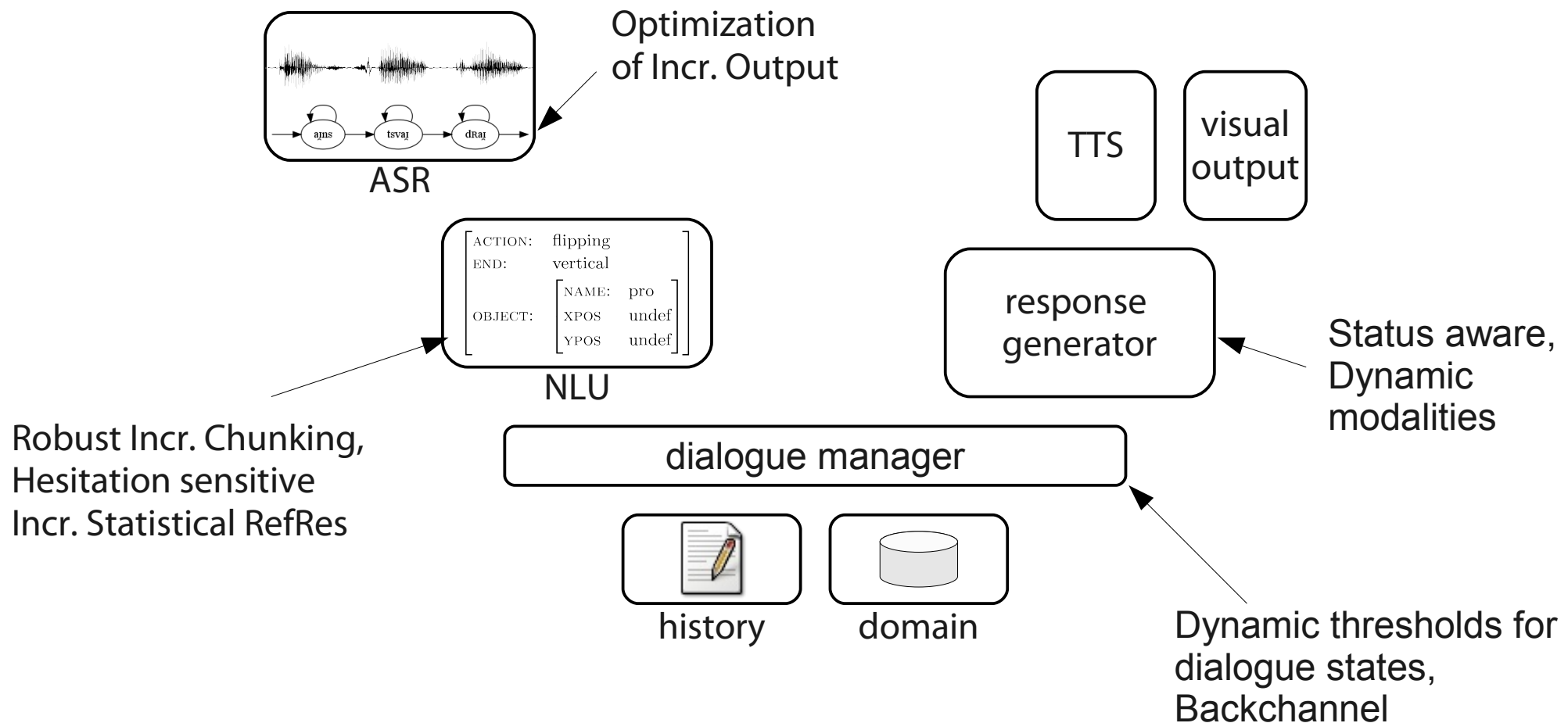
Dialog Management

Action Management

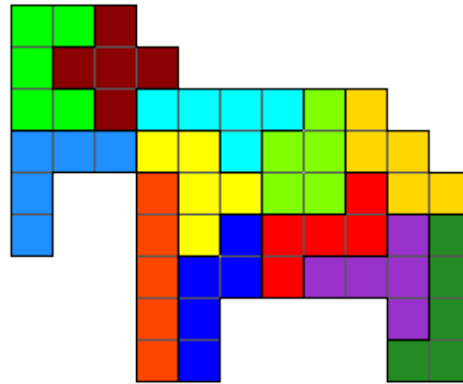
- Action Management delegates output to GUI and speech production modalities.
 - Aware of status of output modalities (busy/idle), can choose free modality if a dialog act can be embodied in either.
-

Summing Up

Incremental component features



Thank You!



Acknowledgements:

David Schlangen

DFG for funding (Emmy Noether programme)

Dialogue Manager

Incremental vs transaction-based DM (3)

Transaction-based timeouts (from VXML)

