Modeling Dialogue

Building Highly Responsive Conversational Agents

David Schlangen, Stefan Kopp
with Sören Klett
CITEC // Bielefeld University

Universität Bielefeld

CITEC
Cognitive Interaction Technology
Center of Excellence
Bielefeld University

Takeaways from Day 1

- Responsive agents: minimize <u>time</u> between *event* and *response,* respond to many more types of events than “end of turn”

- Dialogue participants

    - try to reach mutual understanding

    - continuously monitor whether they have reached it

    - and, if necessary, repair ASAP;

    - so if you don’t react, you risk repair.

- Responsiveness is built into the fabric of dialogue / builds the fabric.

- Reducing it makes (spoken) dialogue *harder*. (Brannigan *et al.* 2011)

Overview of Course

- Day 1: Motivation, Phenomena
- Day 2: Technical Challenges, Approaches
- Day 3: Introduction to Technical Framework
- Day 4: Tasks & Hands-On Exercises
- Day 5: Reports, Discussion

Modeling Dialogue

Building Highly Responsive Conversational Agents

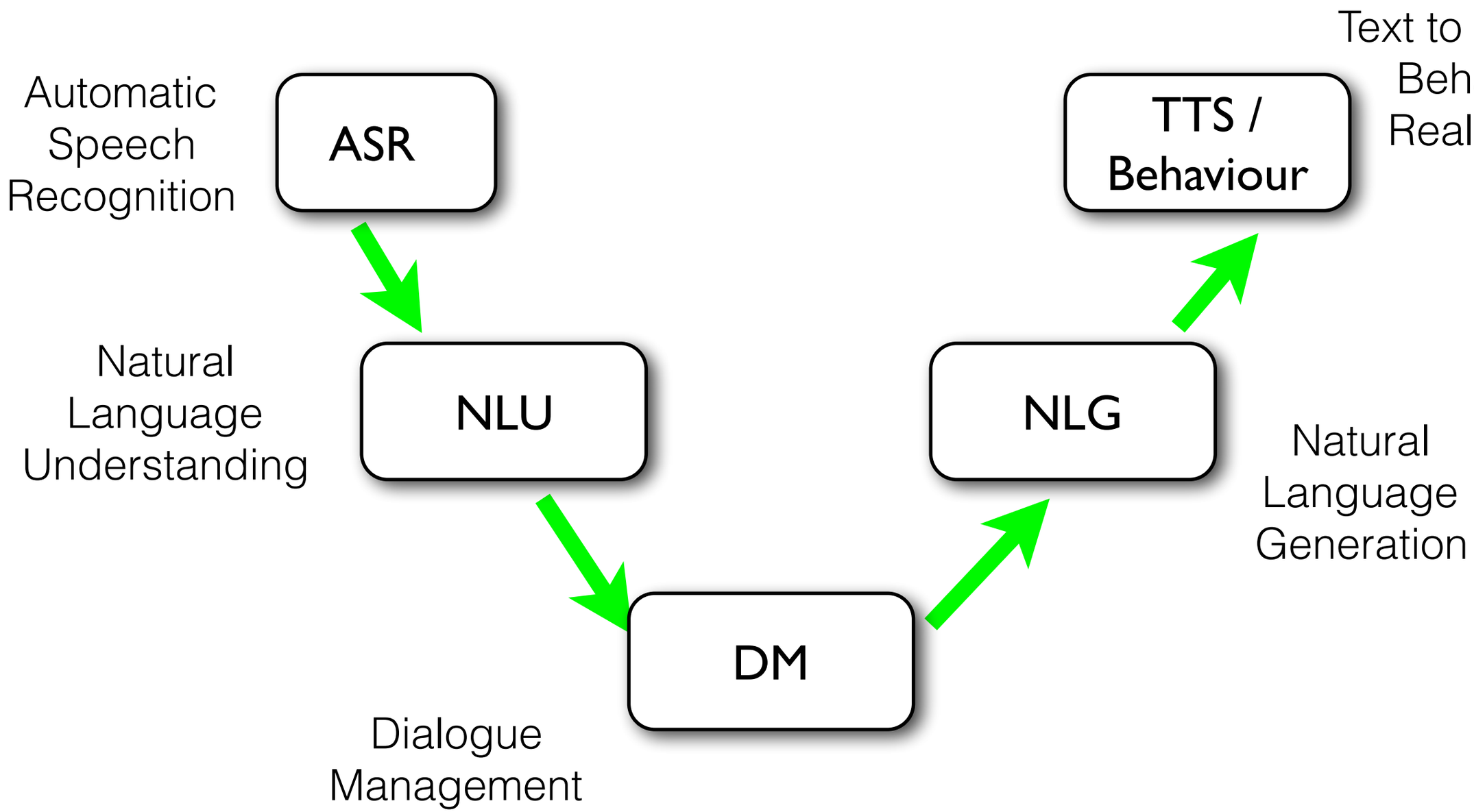
Day 2: Technical Challenges, Approaches

David Schlangen, Stefan Kopp
with Sören Klett
CITEC // Bielefeld University

Universität Bielefeld

CITEC
Cognitive Interaction Technology
Center of Excellence
Bielefeld University

Dialogue System Modules



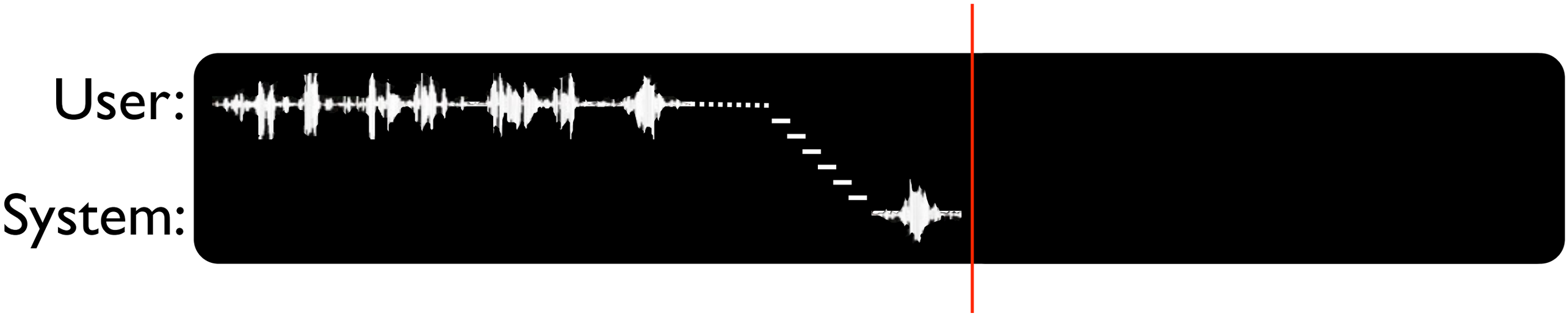
Overview of Day 2

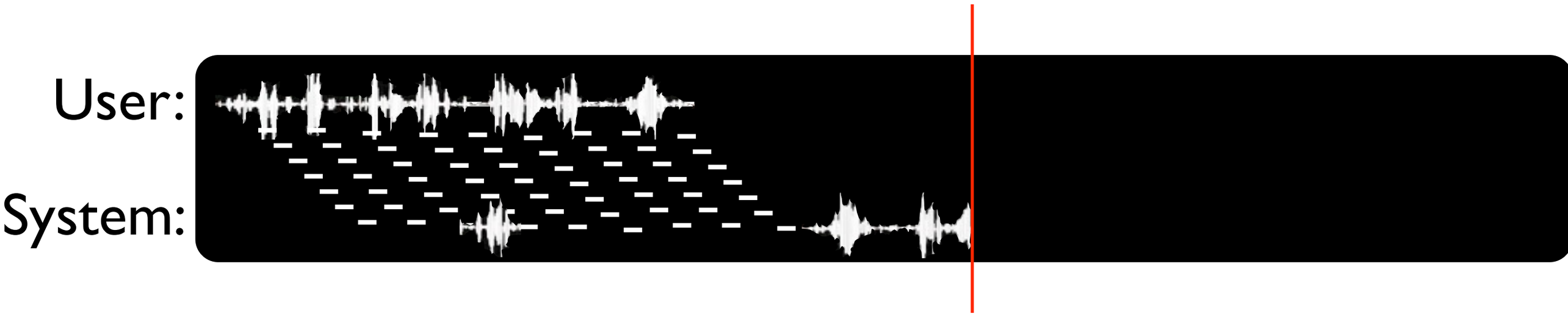
- Information Flow in Incremental Dialogue Processing
- Incremental
  - ASR
  - NLU
  - DM
  - NLG / NVBG
  - Synthesis / Realizer

"Respond to many more types of events than "end of turn"

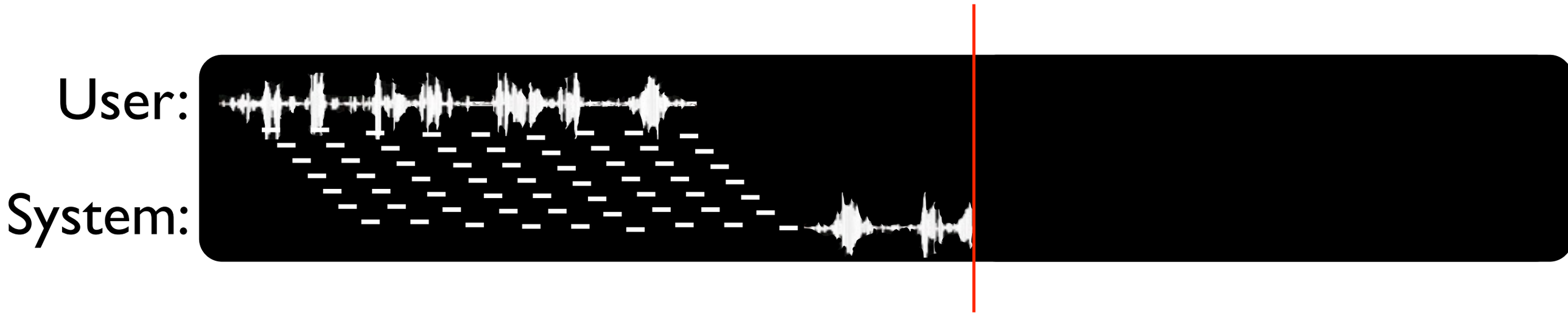
- To do:
  - Create these events
  - Generate appropriate responses.

Non-Incremental vs. Incremental Processing



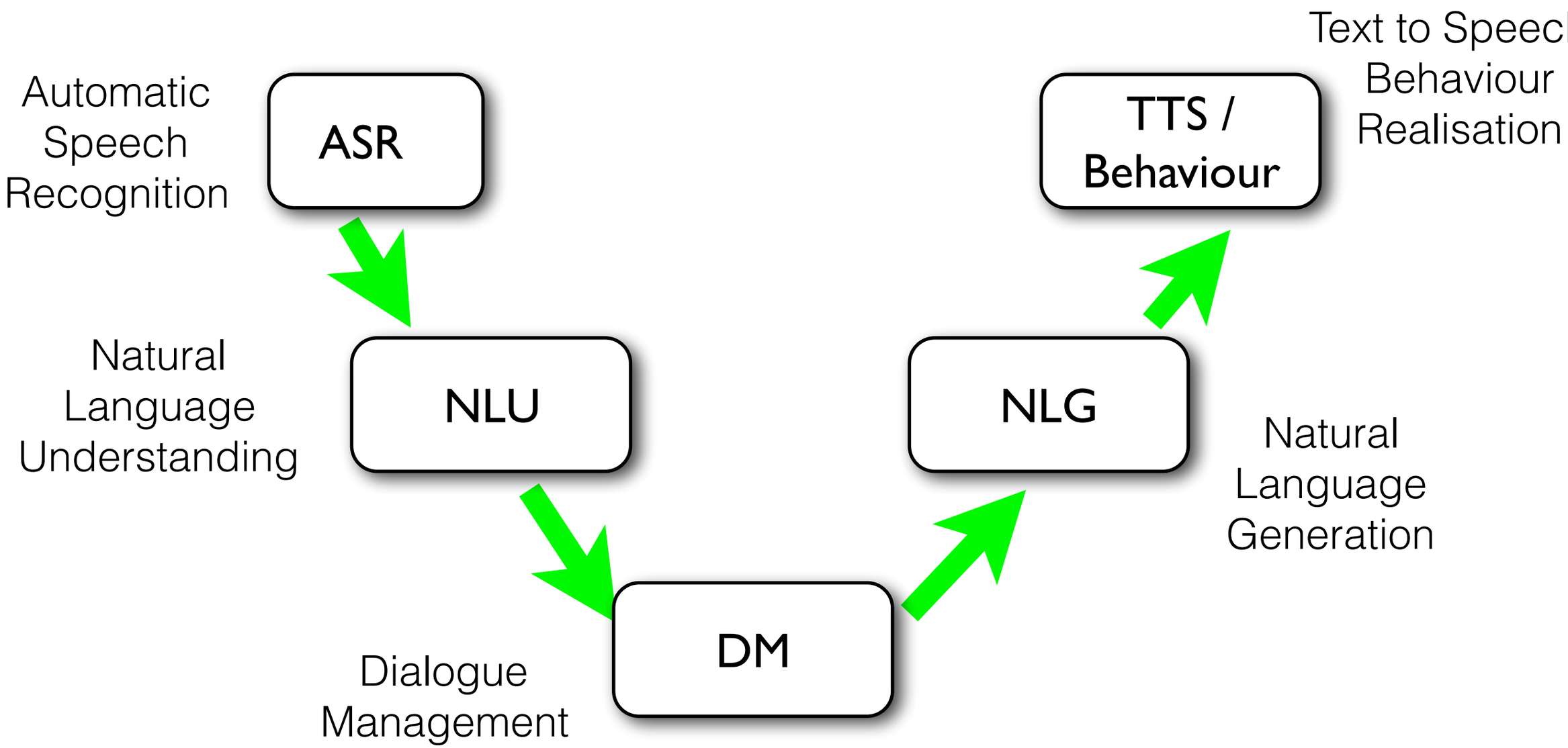


2.1 Challenges



- Requires reconceptualisation of information flow
- Introduces (even more...) uncertainty

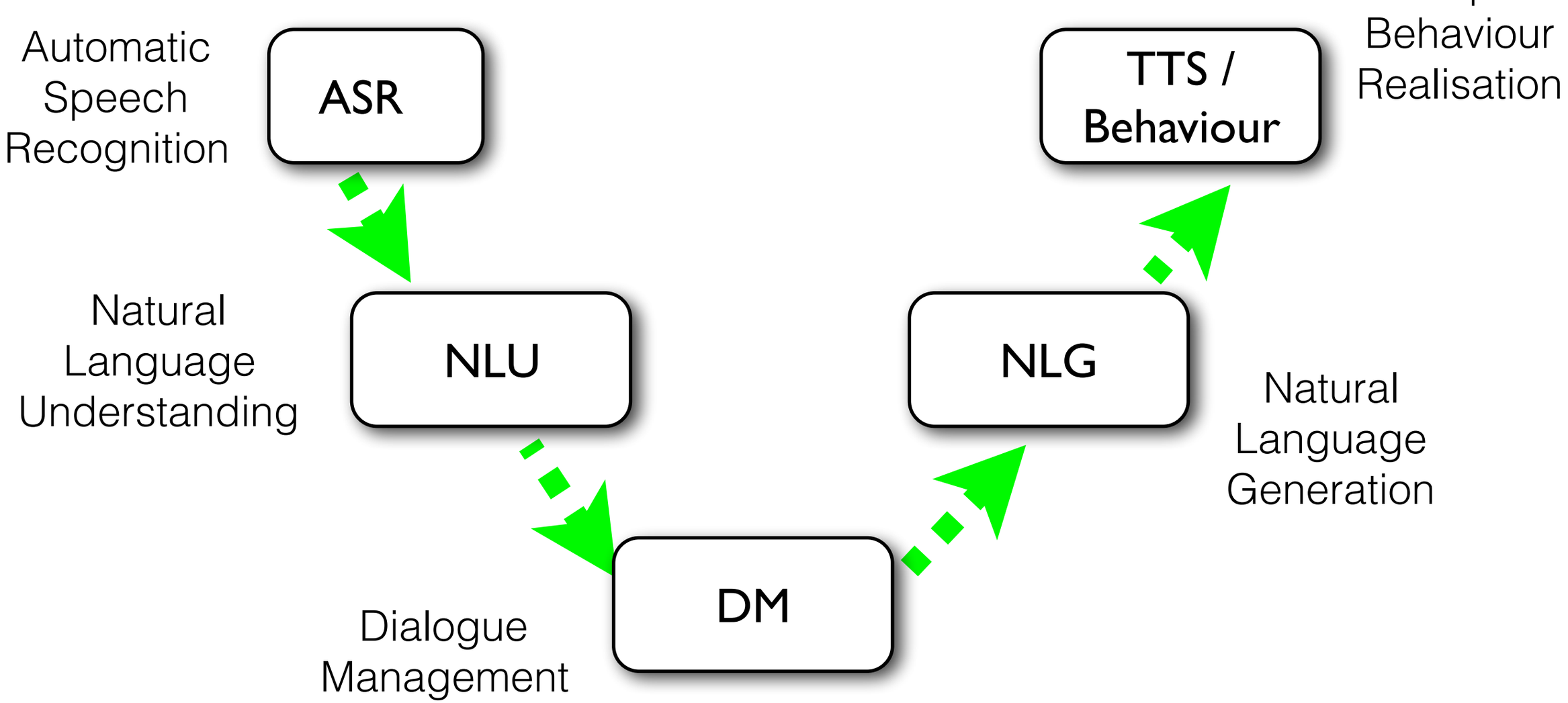
"Incremental Units" model (Schlangen & Skantze EACL 2009, *Dialogue & Discourse* 2011)


Automatic
Speech
Recognition
ASR
Natural
Language
Understanding
NLU
Dialogue
Management
DM
NLG
Natural
Language
Generation
TTS /
Behaviour
Text to Speech /
Behaviour
Realisation

the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised



the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

the IU model

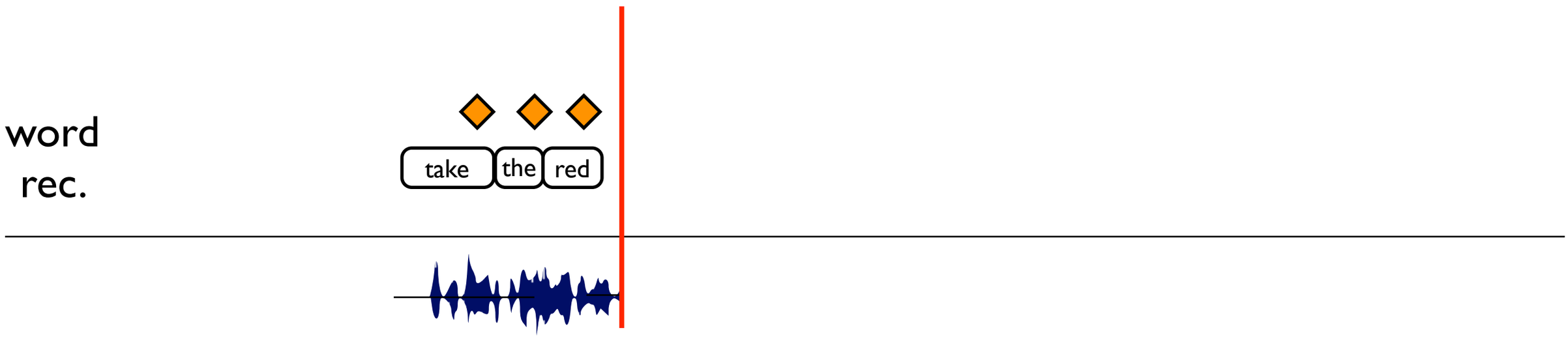
– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

the IU model

– Assumptions –

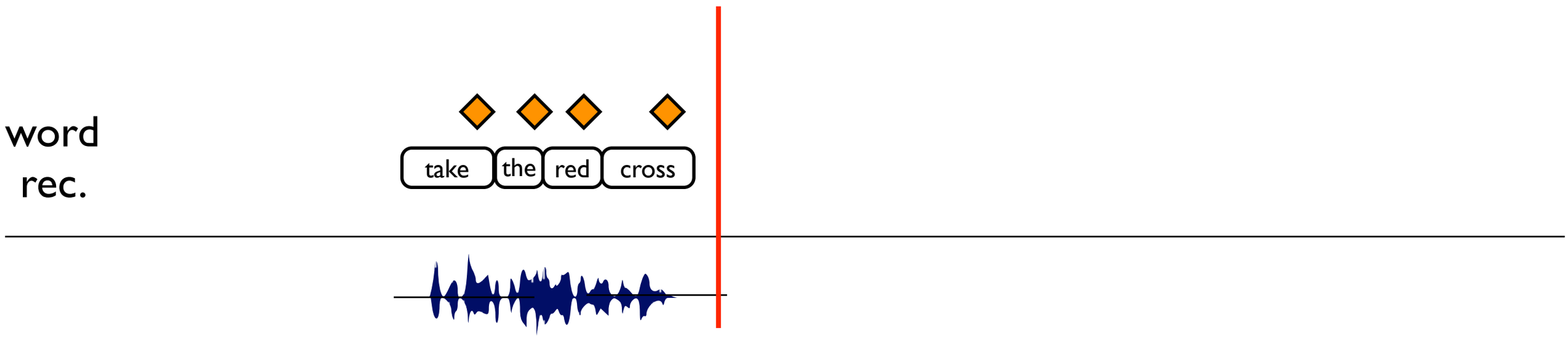
- Information state is updated with *minimal units* of information, as soon as they can be hypothesised



the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

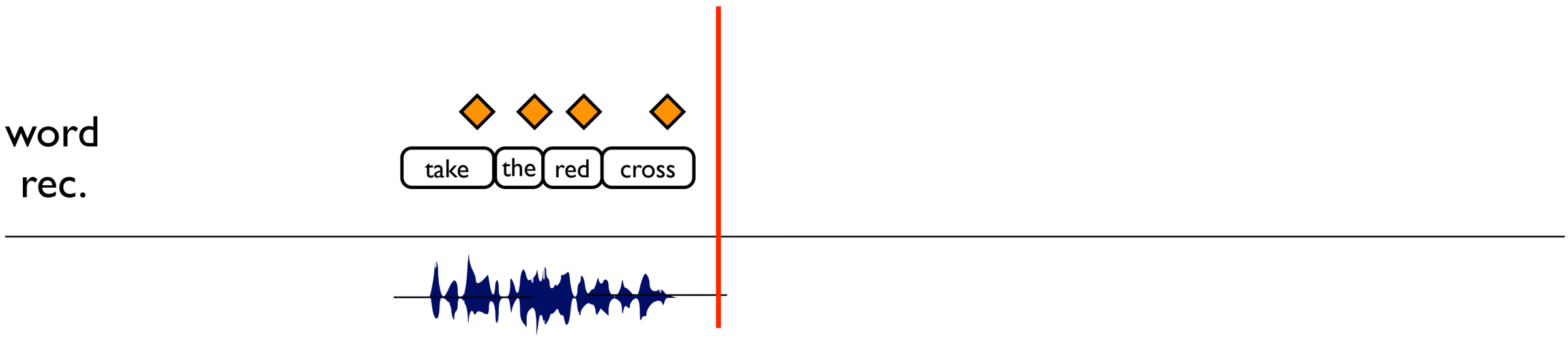


the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

- "Higher-level" hypotheses can be formed on the basis of "lower-level" ones.

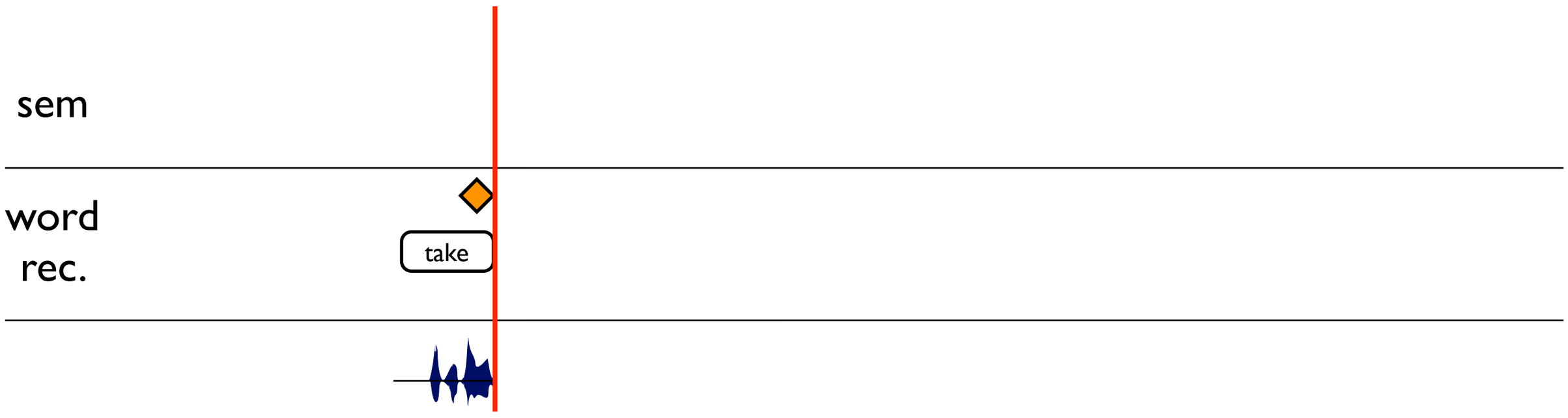


the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

- "Higher-level" hypotheses can be formed on the basis of "lower-level" ones.

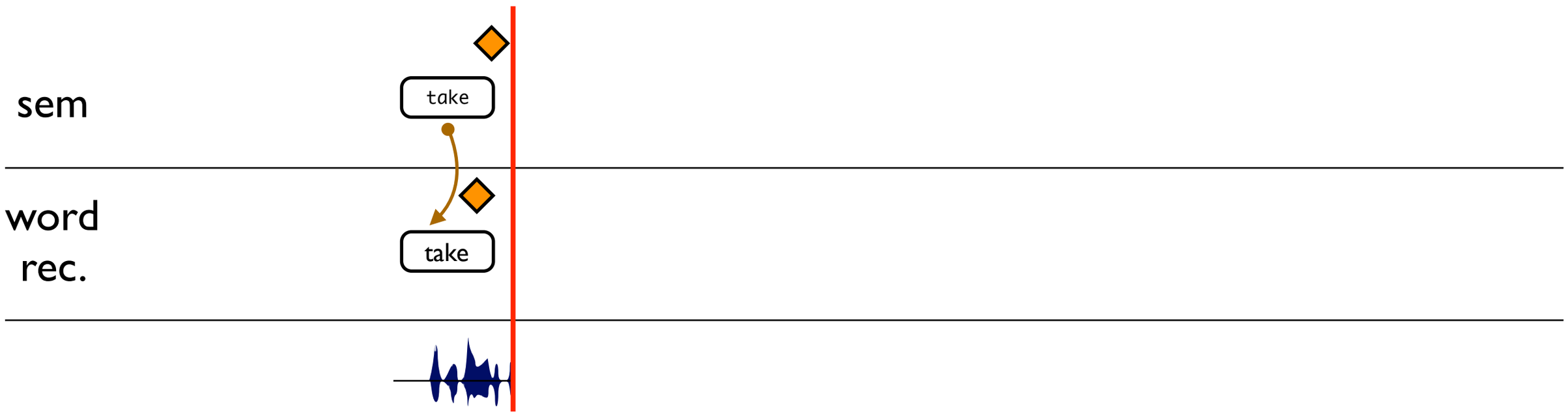


the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

- "Higher-level" hypotheses can be formed on the basis of "lower-level" ones.

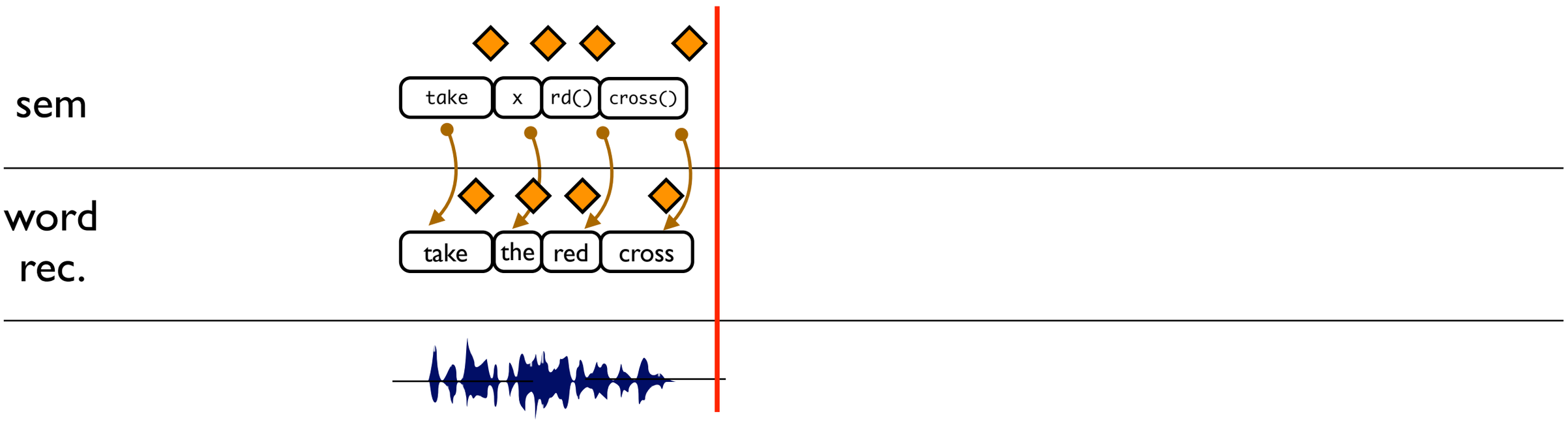


the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

- "Higher-level" hypotheses can be formed on the basis of "lower-level" ones.



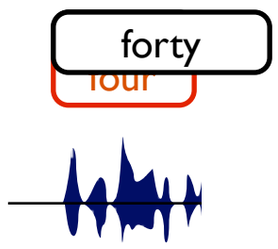
the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

- "Higher-level" hypotheses can be formed on the basis of "lower-level" ones.

- IS may have to be revised, in light of newer information



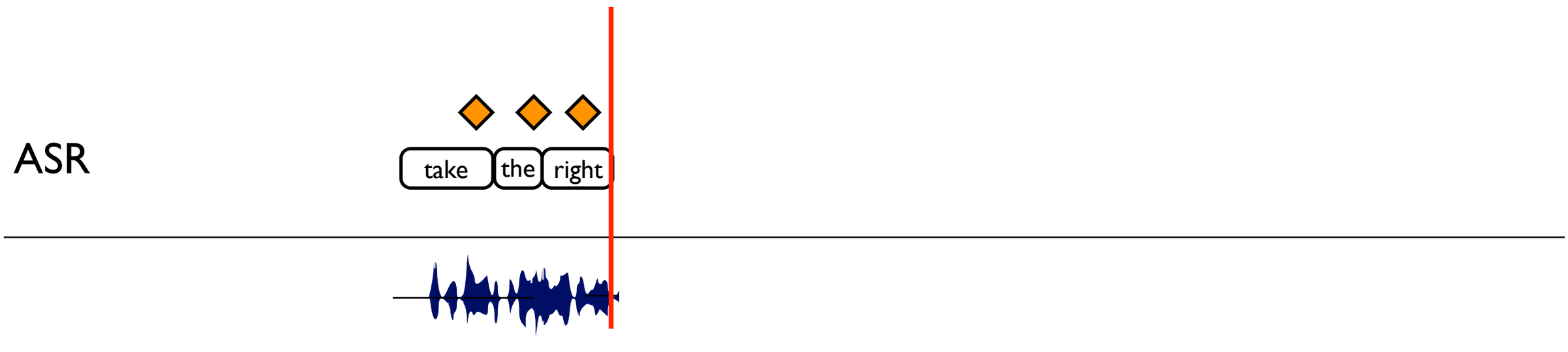
the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

- "Higher-level" hypotheses can be formed on the basis of "lower-level" ones.

- IS may have to be revised, in light of newer information



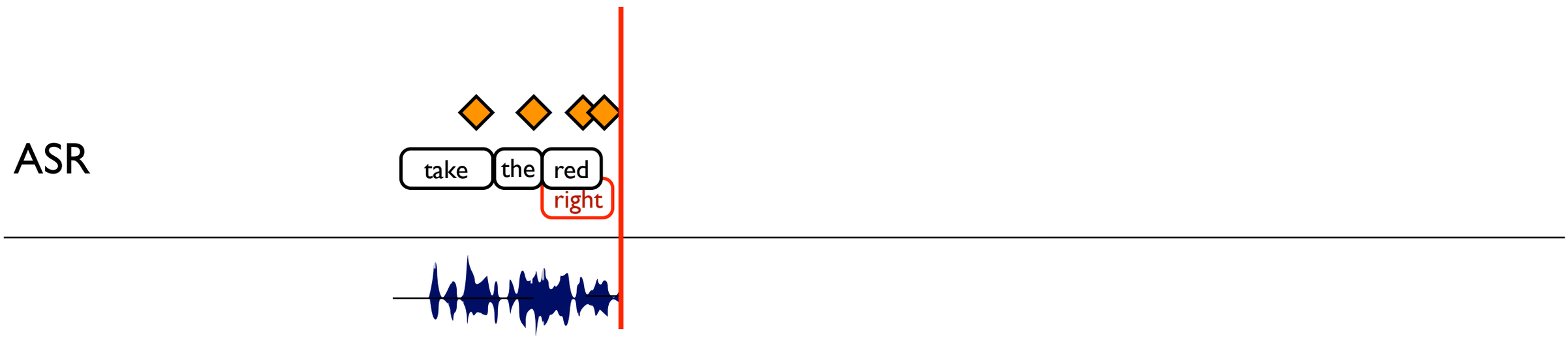
the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

- "Higher-level" hypotheses can be formed on the basis of "lower-level" ones.

- IS may have to be revised, in light of newer information



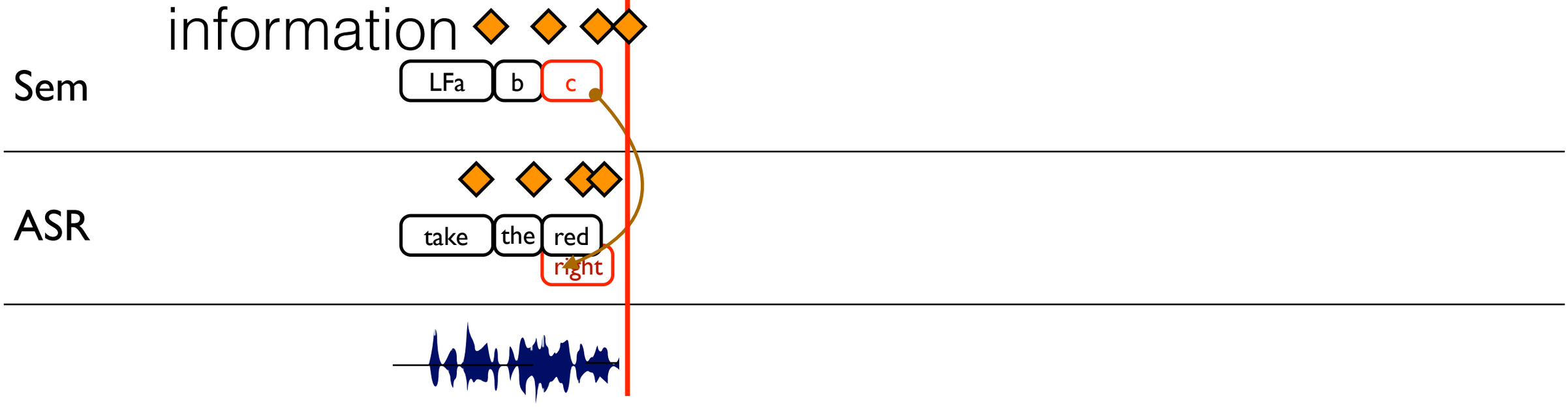
the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

- "Higher-level" hypotheses can be formed on the basis of "lower-level" ones.

- IS may have to be revised, in light of newer information

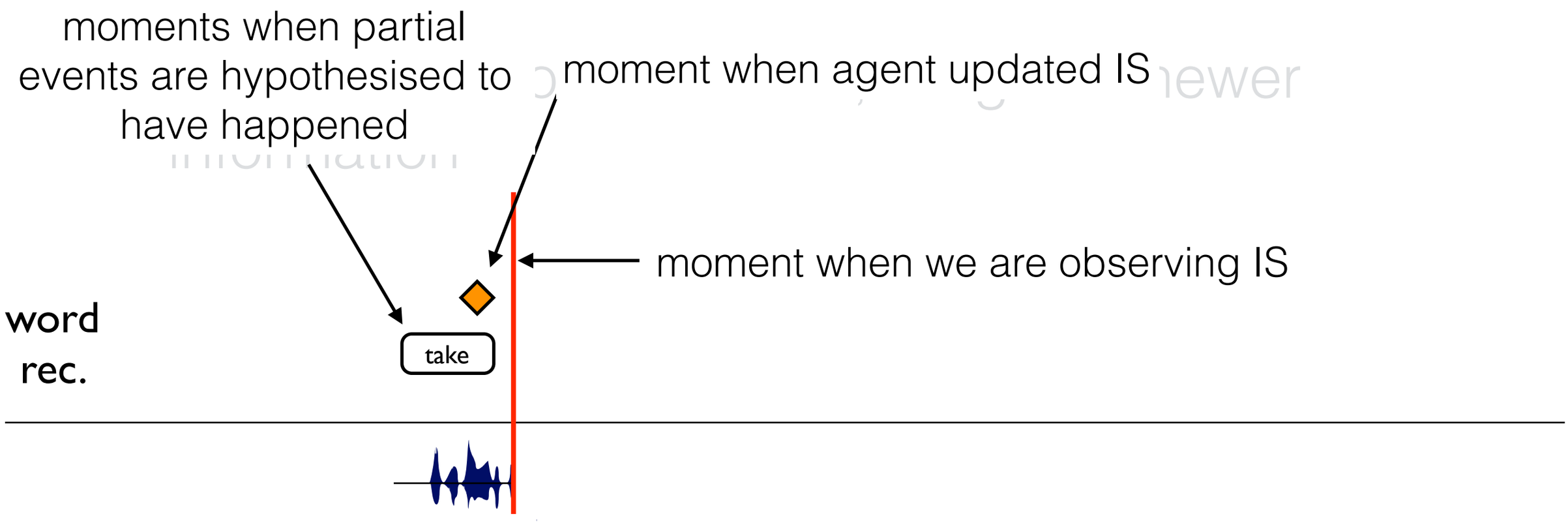


the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

- "Higher-level" hypotheses can be formed on the basis of "lower-level" ones.



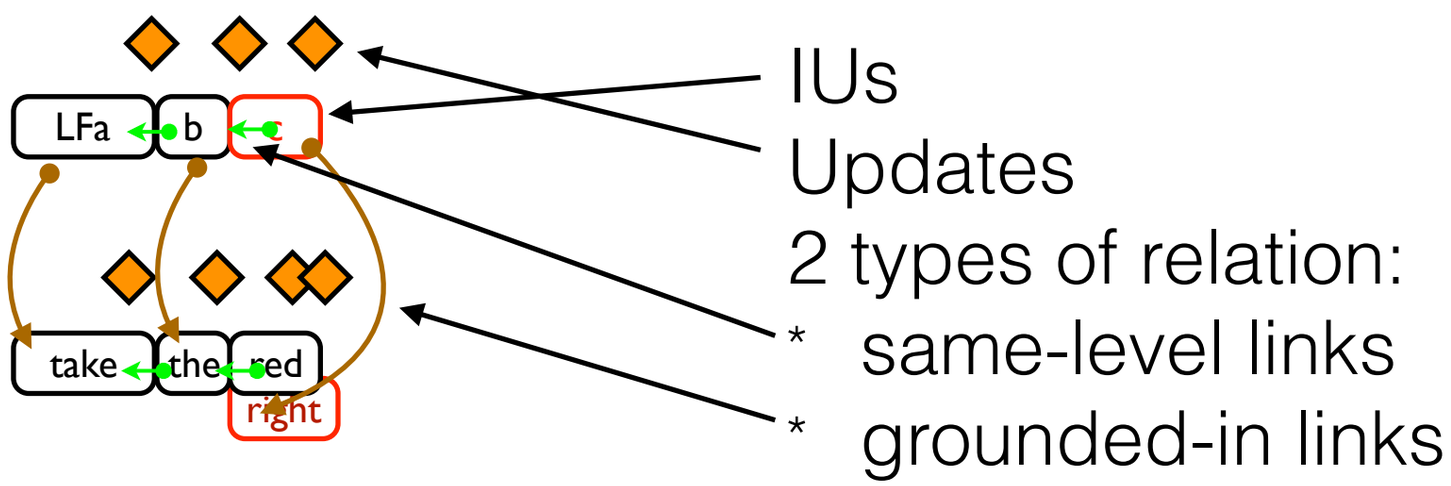
the IU model

– Assumptions –

- Information state is updated with *minimal units* of information, as soon as they can be hypothesised

- "Higher-level" hypotheses can be formed on the basis of "lower-level" ones.

- IS may have to be revised, in light of newer information



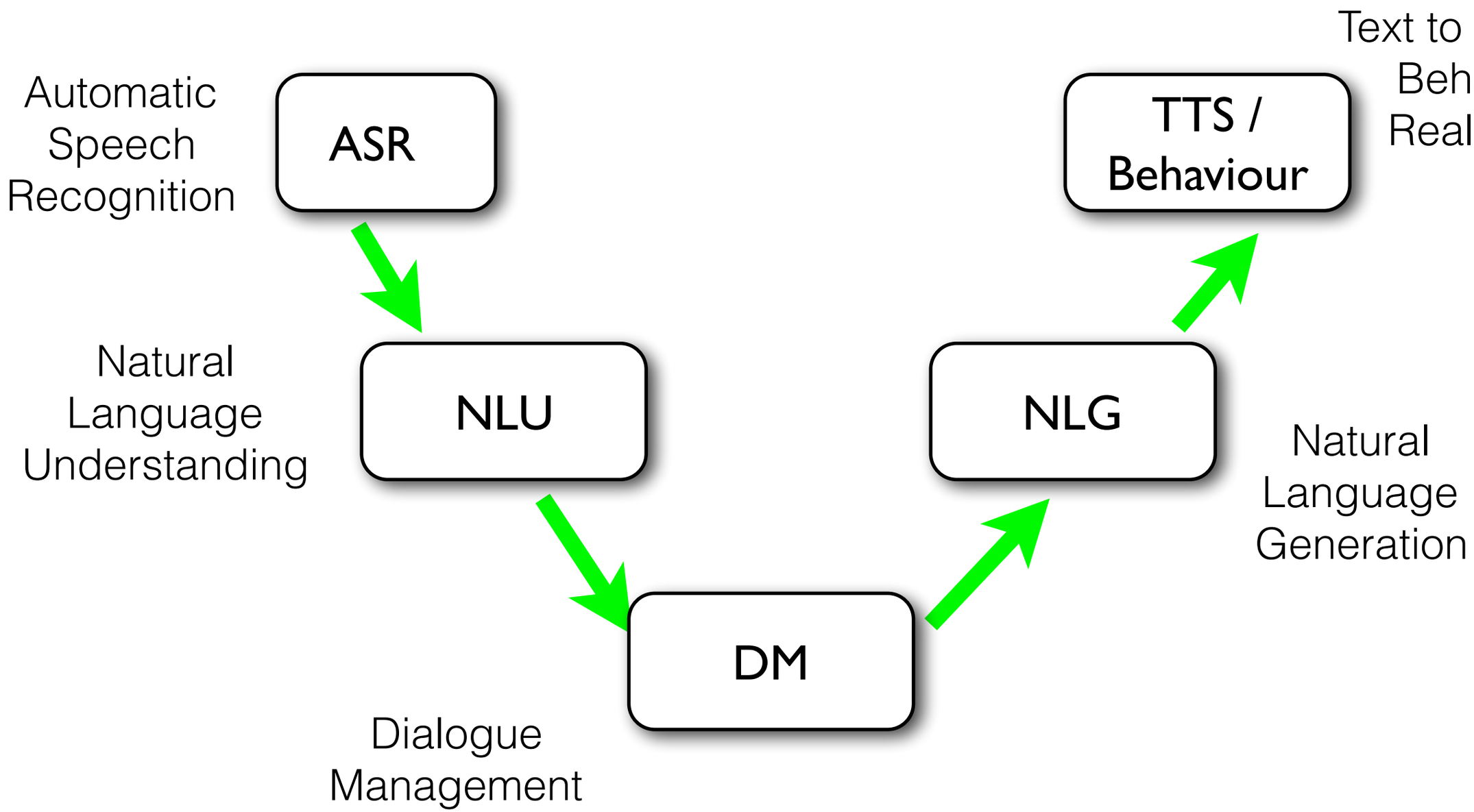
the IU model

- Implemented in InproTK (http://www.inpro.tk), Jindigo (Skantze), IPAACA (Kopp & Buschmeier), HiRAF (Klett *et al.*)

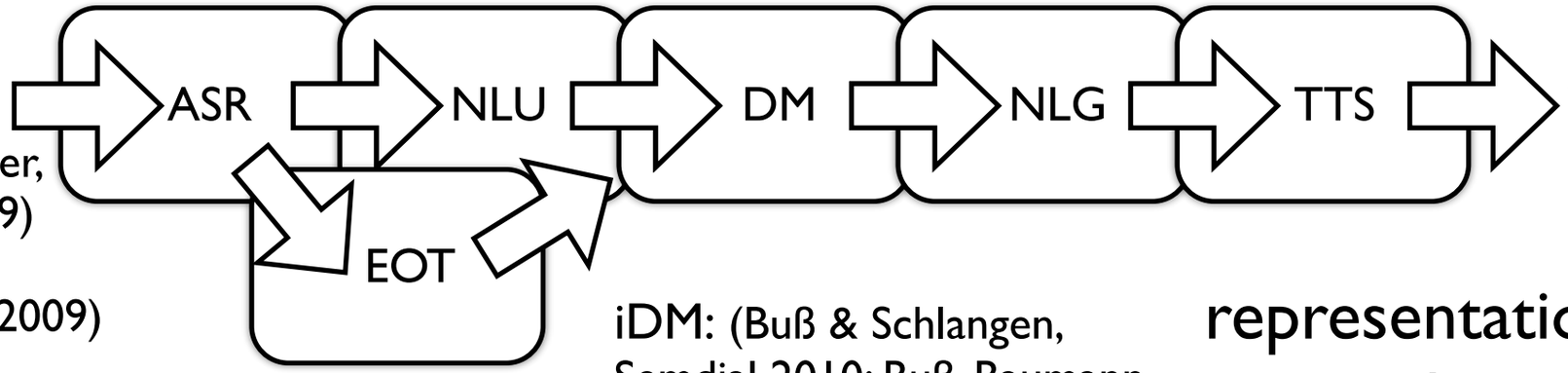
Overview of Day 2

- Information Flow in Incremental Dialogue Processing
- Incremental
  - ASR
  - NLU
  - DM
  - NLG / NVBG
  - Synthesis / Realizer

Dialogue System Modules



completion: (Baumann & Schlangen, SIGdial 2011)



(Baumann & Schlangen, ACL demo 2012, Interspeech 2012; Buschmeier et al. SIGdial 2012)

iASR: (Baumann, Atterer, Schlangen; NAACL 2009) (Baumann, Buß, Atterer, Schlangen; Interspeech 2009)

iDM: (Buß & Schlangen, Semdial 2010; Buß, Baumann, Schlangen, SIGdial 2010; Buß & Schlangen, Semdial 2011)

representations of partial results?

iNLU: (Atterer, Baumann, Schlangen, Interspeech 2009) (Atterer & Schlangen, SRSL 2009) (Schlangen, Baumann, Atterer, SIGdial 2009) (Heintze, Baumann, Schlangen, SIGdial 2010) (Peldszus, Buß, Baumann, Schlangen, EACL 2012)

mechanisms for computing them?

iEOT: (Schlangen, Interspeech 2006), (Baumann; ESSLLI 2008), (Atterer, Baumann, Schlangen, Coling 2009)

evaluation: (Baumann, Buß, Schlangen, D&D 2011)

evaluation?

configurations, interactions?

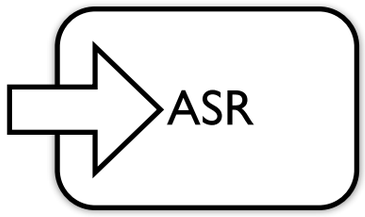
AGMo, impl.: (Schlangen & Skantze, EACL 2009, D&D 2011), (Schlangen et al., SIGdial 2010)

architectures?

Systems: (Schlangen & Skantze, EACL 2009) (Buß & Schlangen Semdial 2010, 2011)

systems?

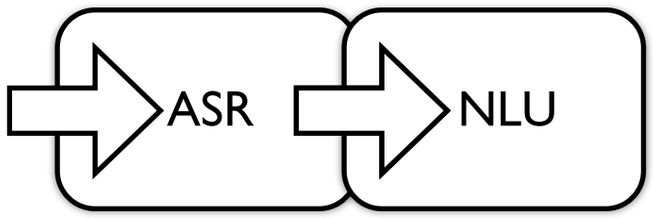
annotated bibliography:
http://www.inpro.tk
(see also http://www.dsg-bielefeld.de)



Part II
Challenges and Approaches
2.2 iASR

ASR creates a lot of instability on the right frontier.
Tradeoff between stability and latency.

(See (Bauman *et al.* 2009 ff.), http://inpro.tk)



Part II

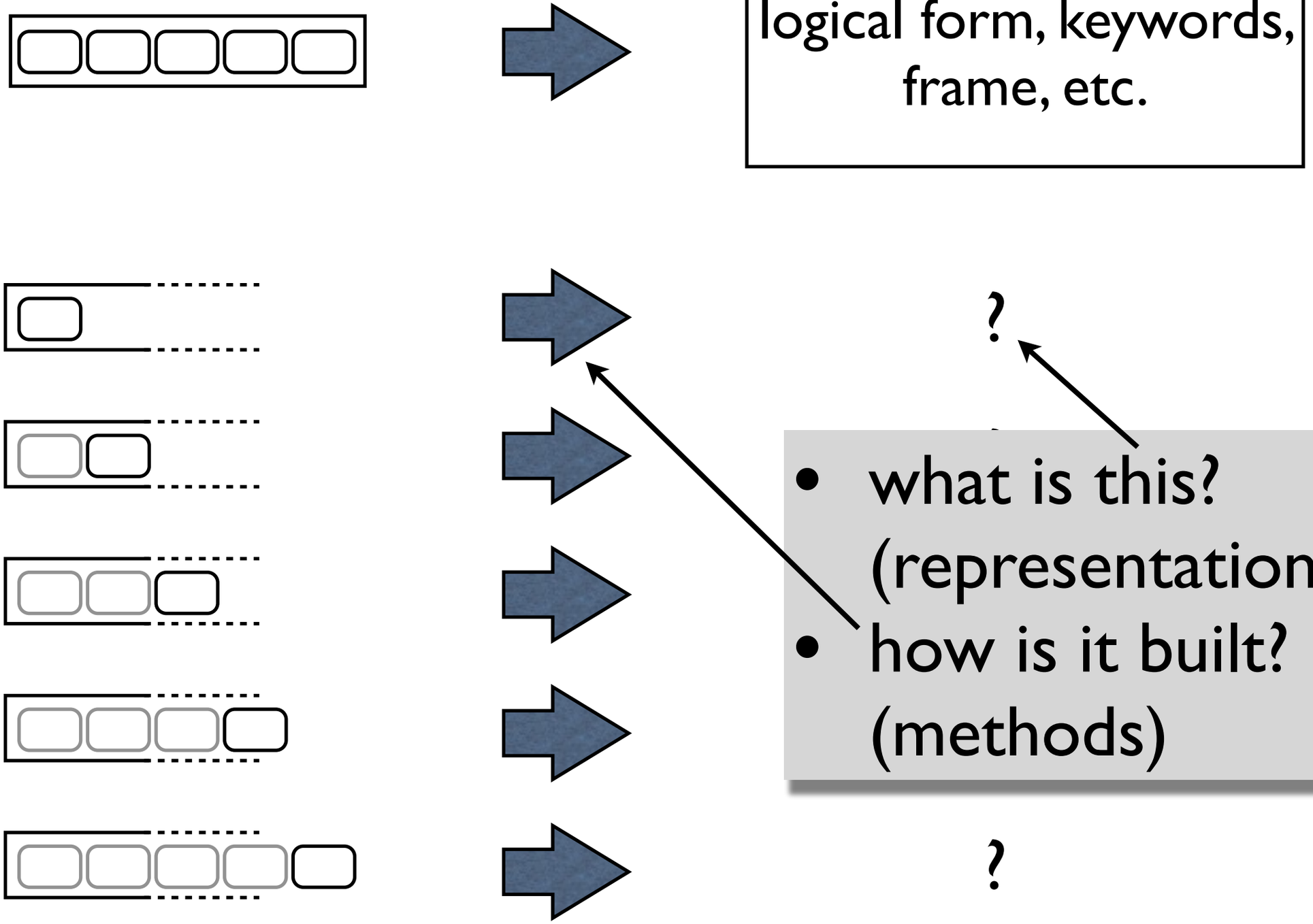
Challenges and Approaches

2.2 iNLU

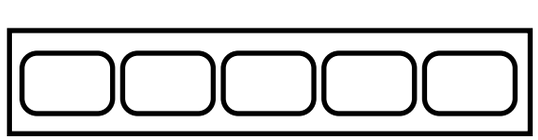
(incremental) NLU

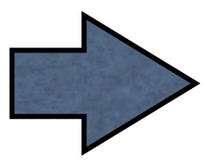
- input: utterances
- output: meaning representations
- the task: extract (intended) meaning from utterance
- incremental: input and/or output are IUs; input IUs are indvidual words output IUs are ?

(incremental) NLU

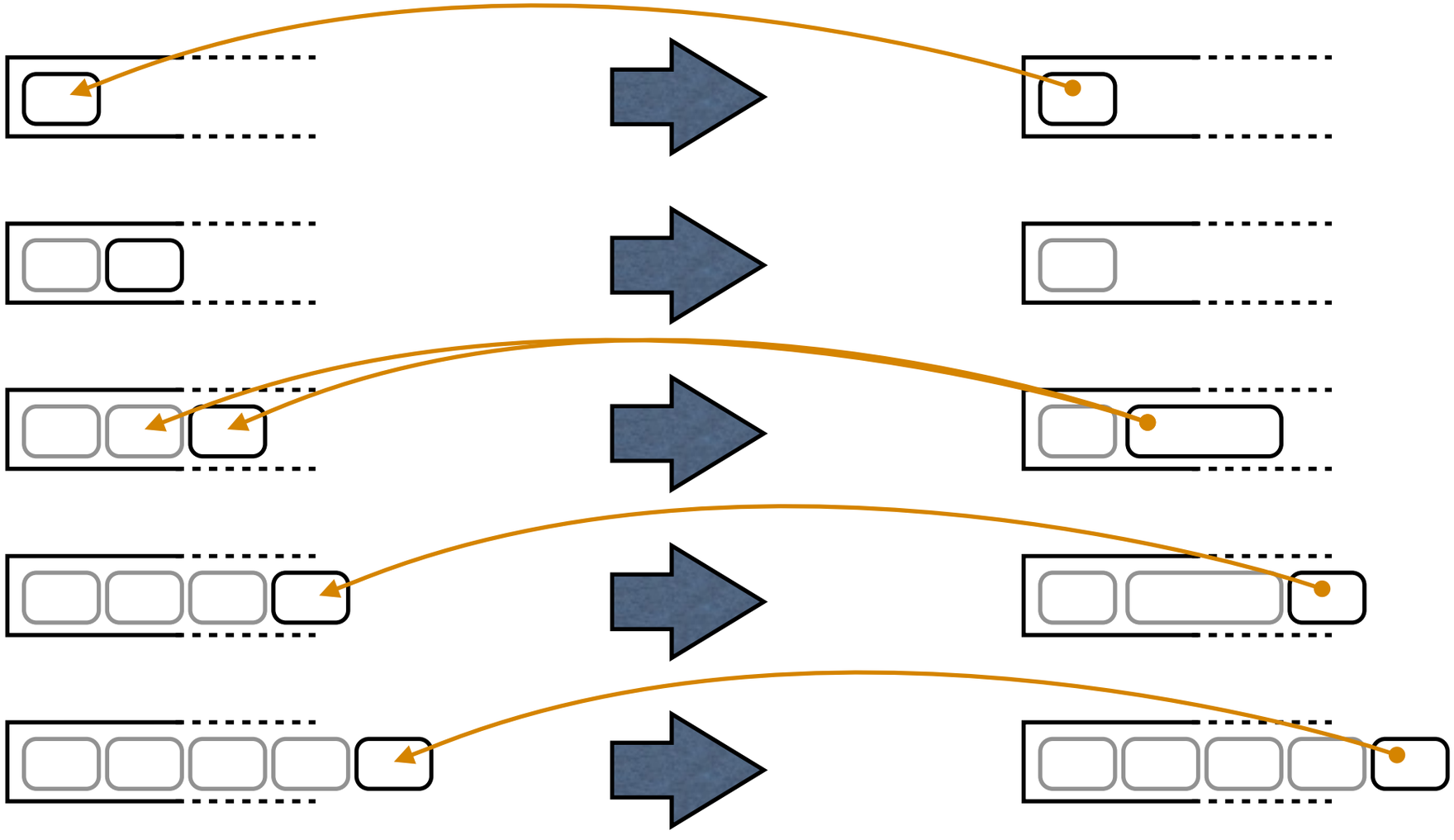


incremental NLU

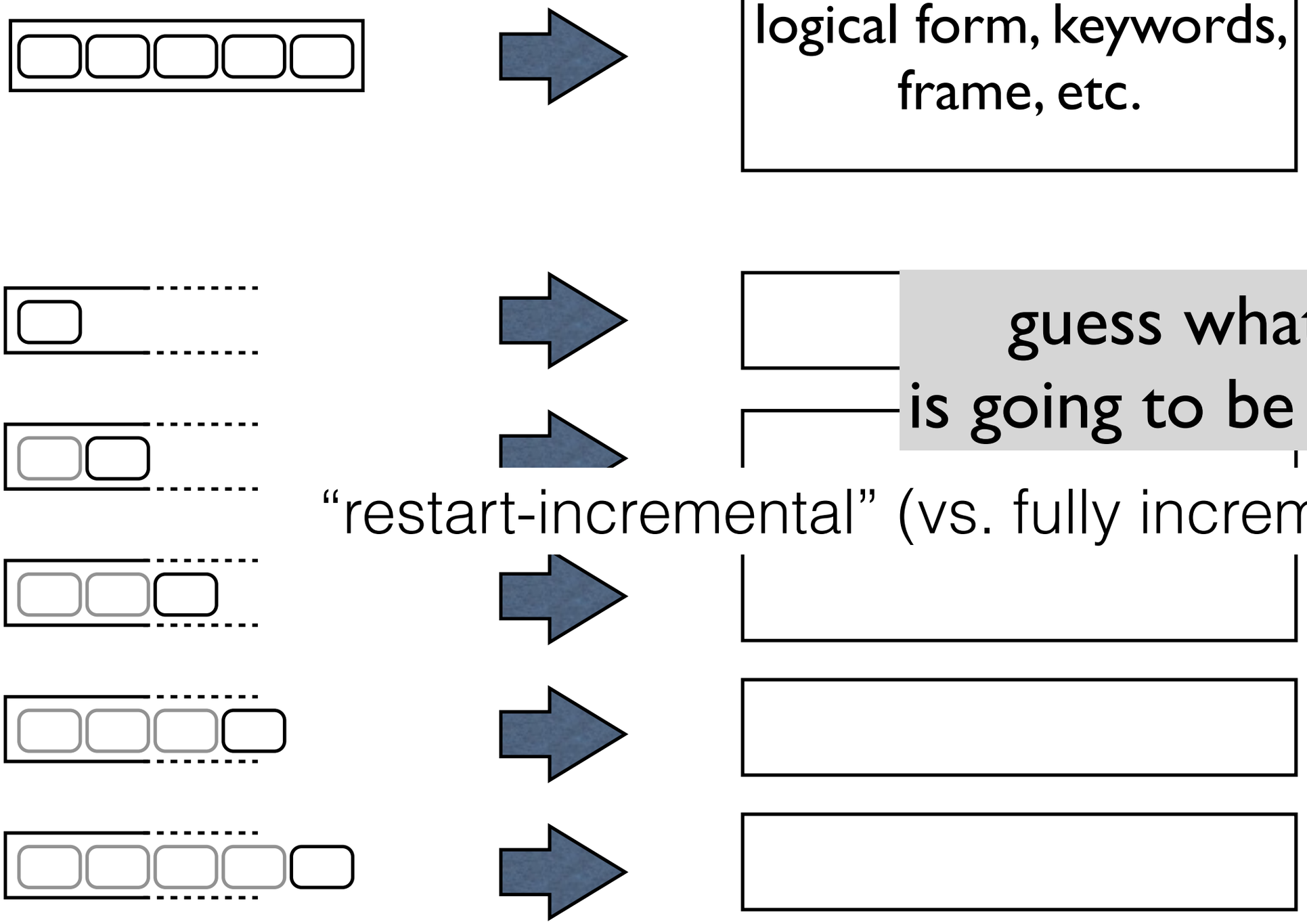




logical form, keywords, frame, etc.

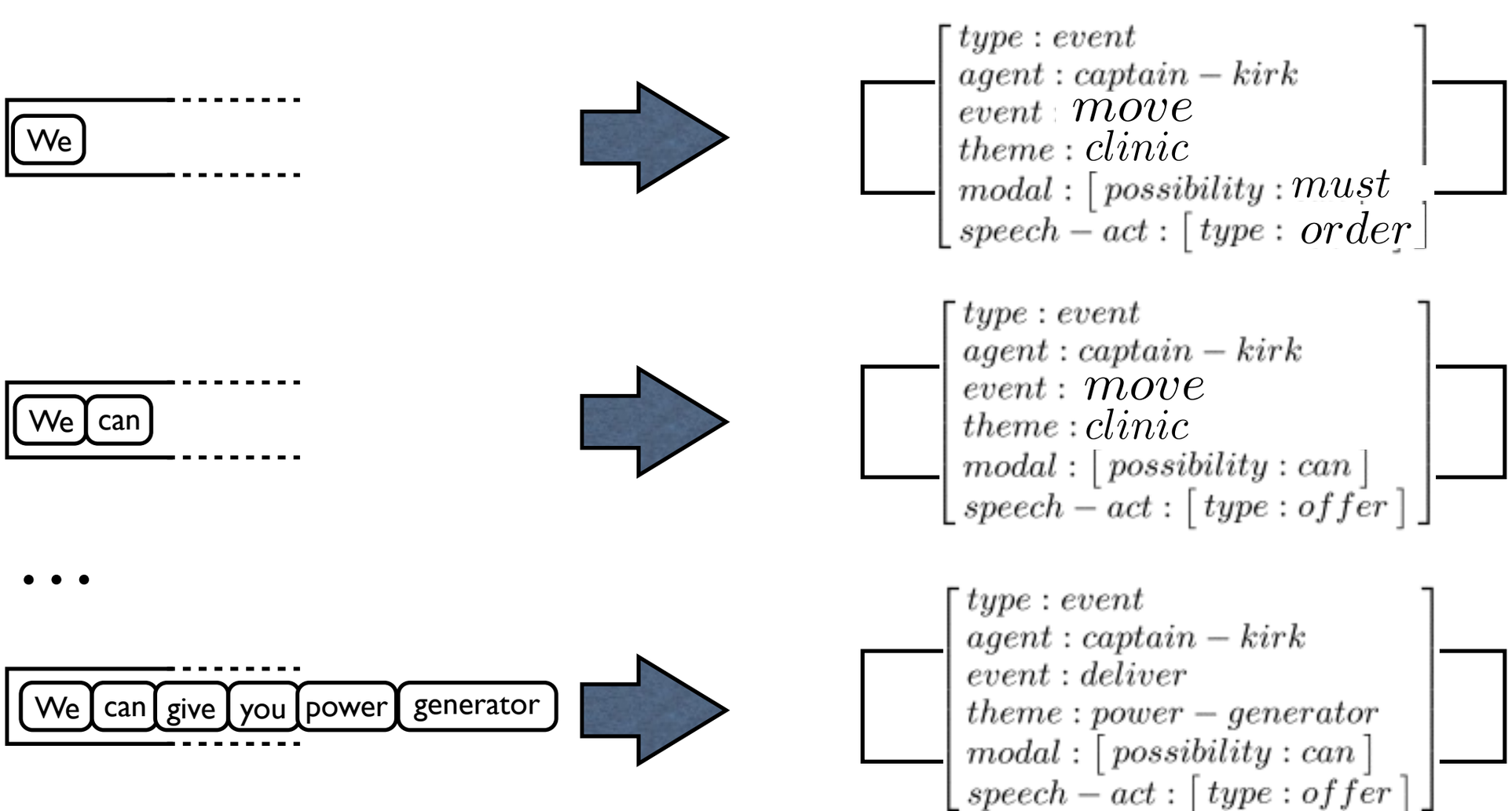


incremental NLU



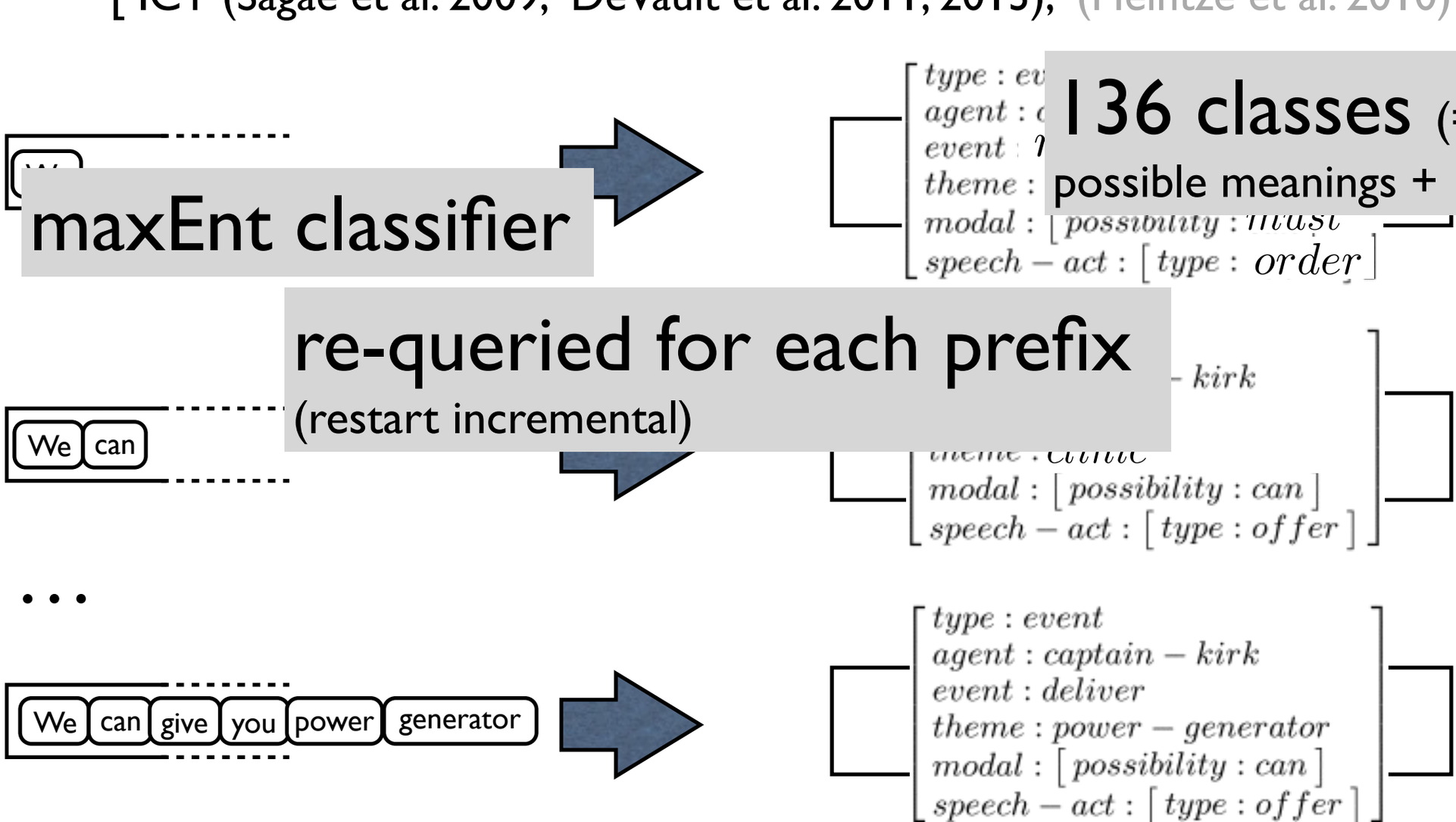
what has been tried?

- predict whole representation: one (massively) multi-class problem
  [ICT (Sagae et al. 2009, DeVault et al. 2011, 2013), (Heintze et al. 2010)



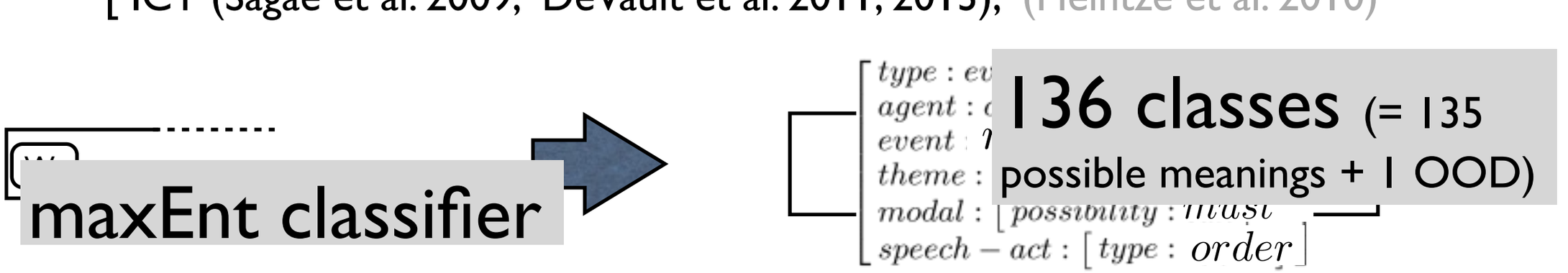
what has been tried?

- predict whole representation: one (massively) multi-class problem
  [ICT (Sagae et al. 2009, DeVault et al. 2011, 2013), (Heintze et al. 2010)



what has been tried?

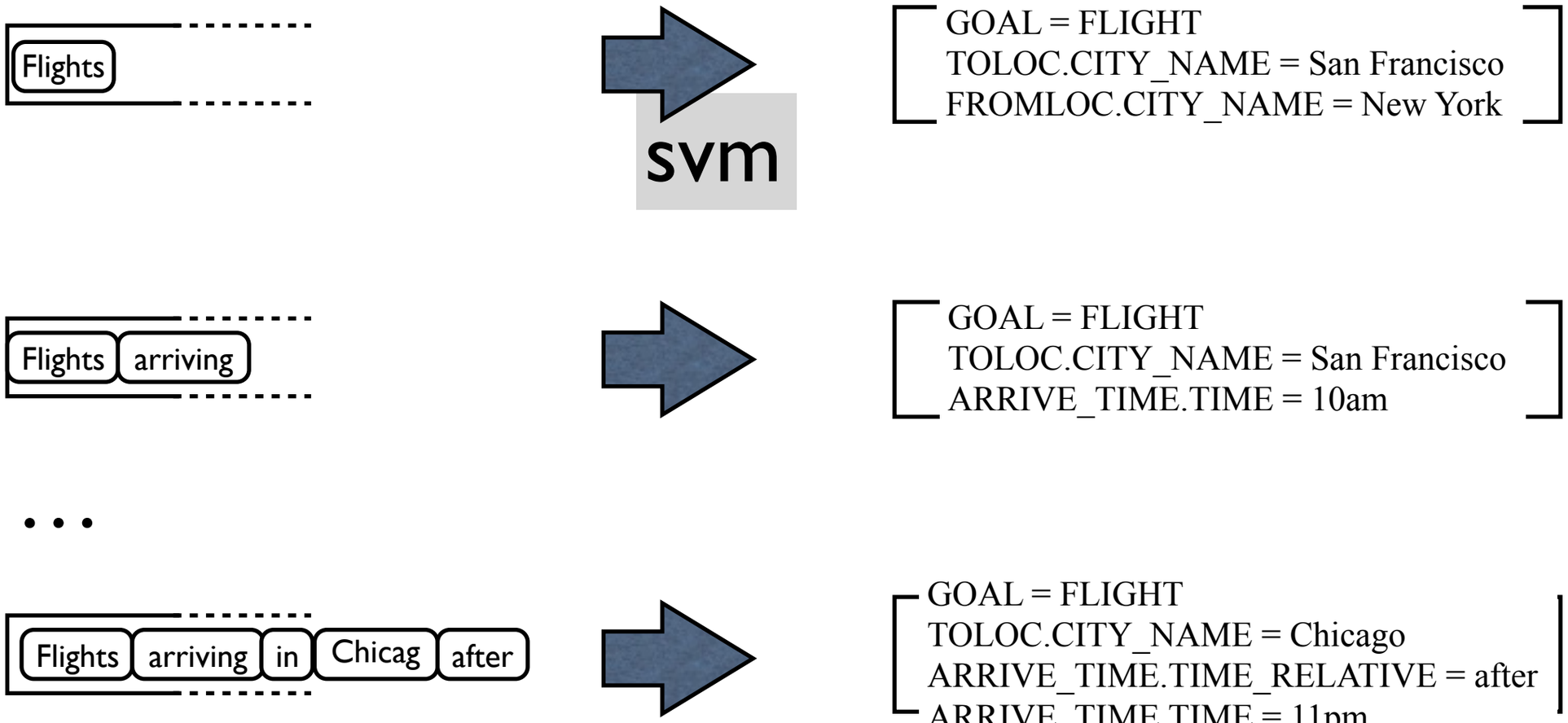
- predict whole representation: one (massively) multi-class problem
  [ICT (Sagae et al. 2009, DeVault et al. 2011, 2013), (Heintze et al. 2010)



what has been tried?

- predict whole representation: one (massively) multi-class problem

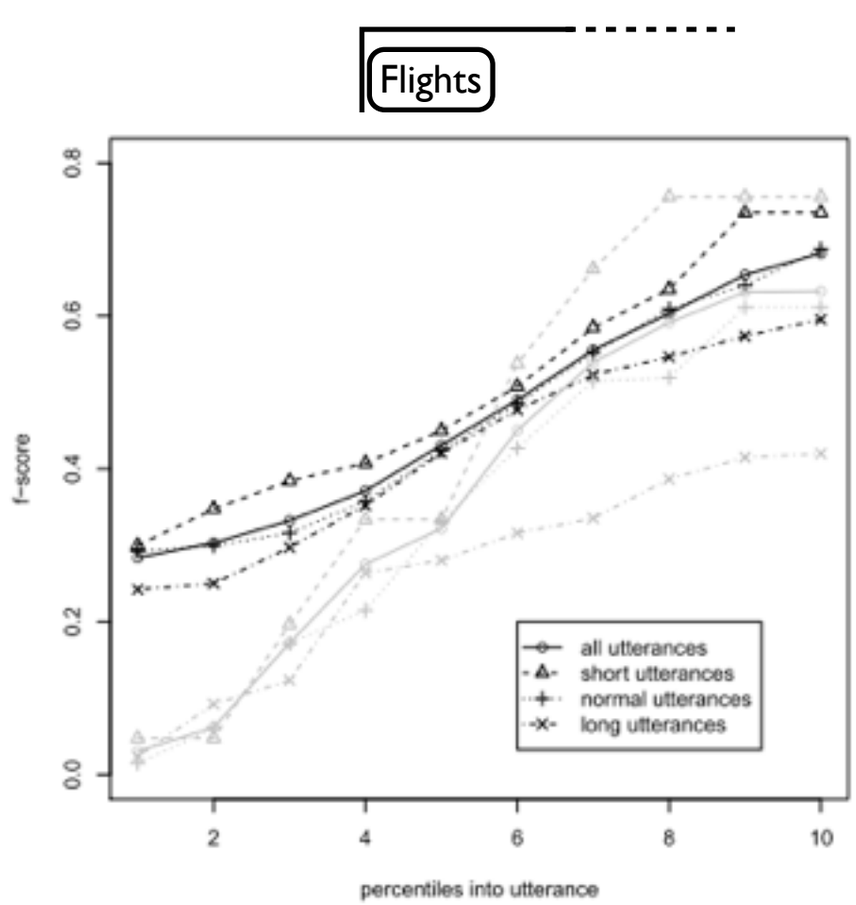
  [ICT (Sagae et al. 2009, DeVault et al. 2011, 2013), (Heintze et al. 2010)



what has been tried?

- predict whole representation: one (massively) multi-class problem

  [ICT (Sagae et al. 2009, DeVault et al. 2011, 2013), (Heintze et al. 2010)



svm

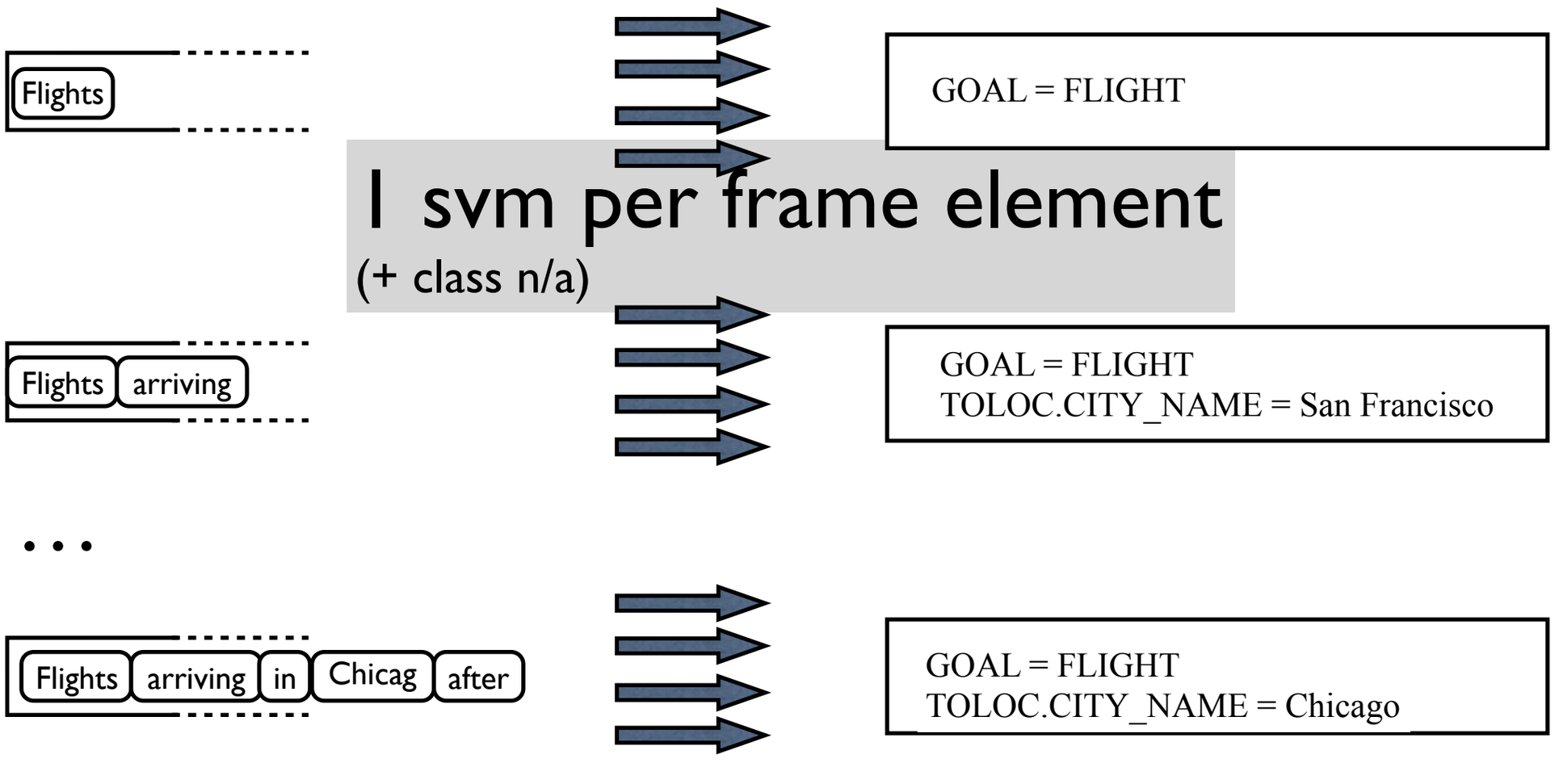
GOAL = FLIGHT
TOLOC.CITY_NAME = San Francisco
FROMLOC.CITY_NAME = New York

3159 classes (!!)
(2594 of which occur only once!)
The curse of combinatorics..
|"From x to y"| = (#Cities)^2

not a good domain for guessing final meaning

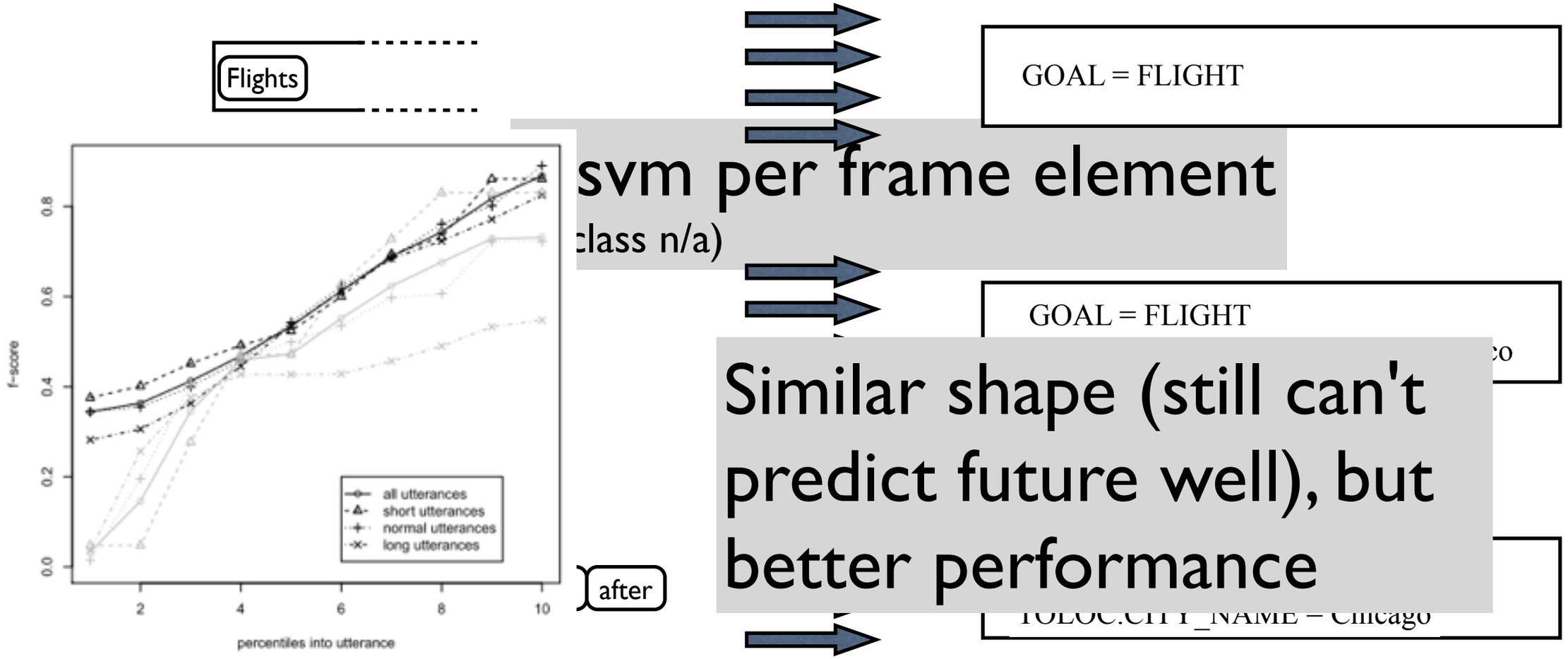
what has been tried?

- separate classifiers for each slot (semi-aligned representation) [(Heintze et al. 2010)]



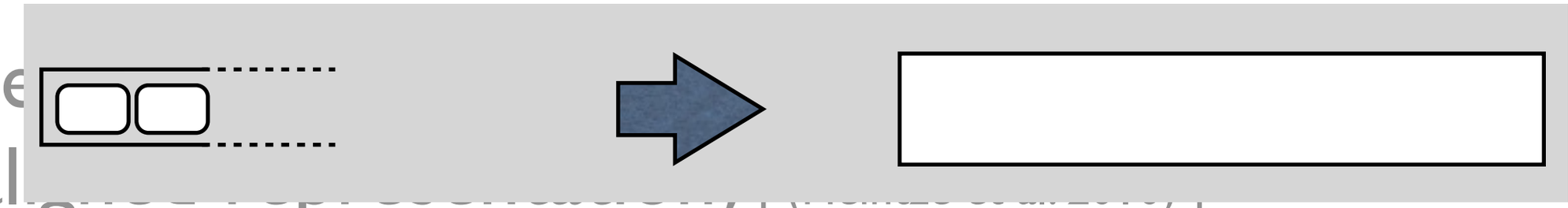
what has been tried?

- separate classifiers for each slot (semi-aligned representation) [(Heintze et al. 2010)]

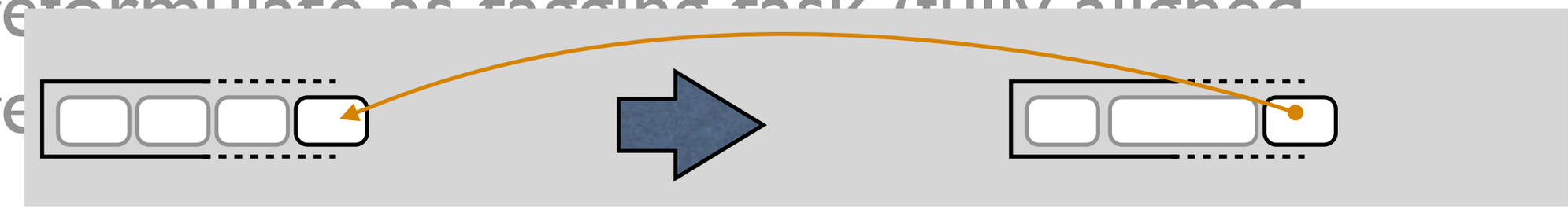


what has been tried?

- predict whole representation: one (massively) multi-class problem
  [ICT (Sagae et al. 2009, DeVault et al. 2011, 2013), (Heintze et al. 2010)]



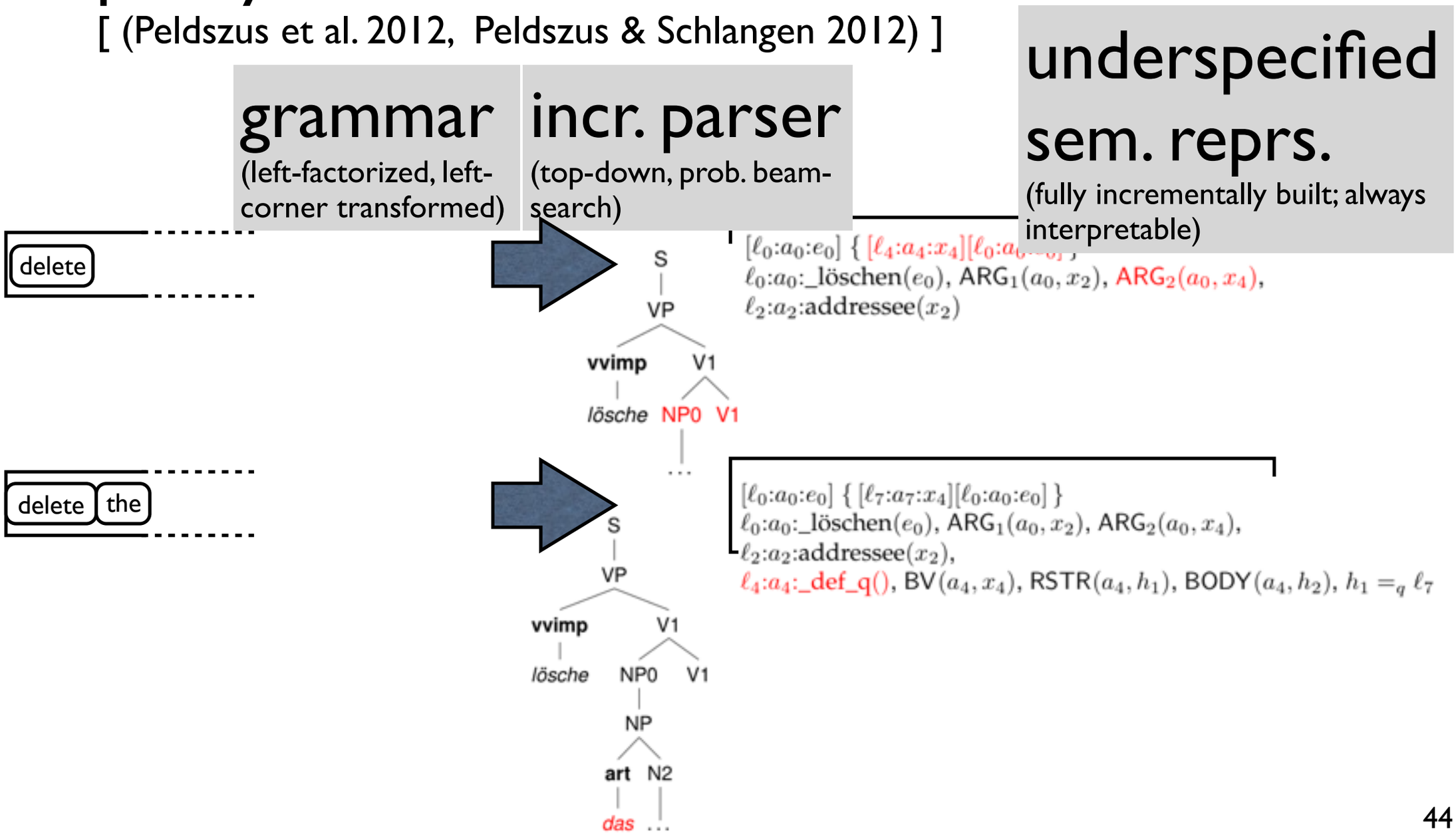
- se
  al
- re
  re



- purely incremental semantics contruction
  [(Peldszus et al. 2012, Peldszus & Schlangen 2012)]

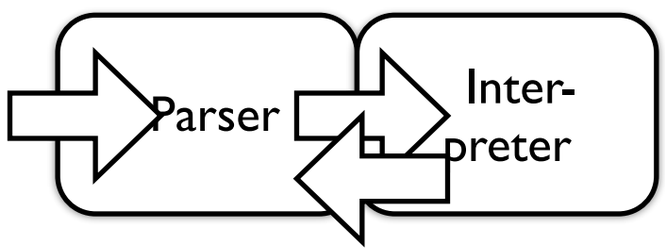
what has been tried?

- purely incremental semantics contruction
  [(Peldszus et al. 2012, Peldszus & Schlangen 2012)]



what has been tried?

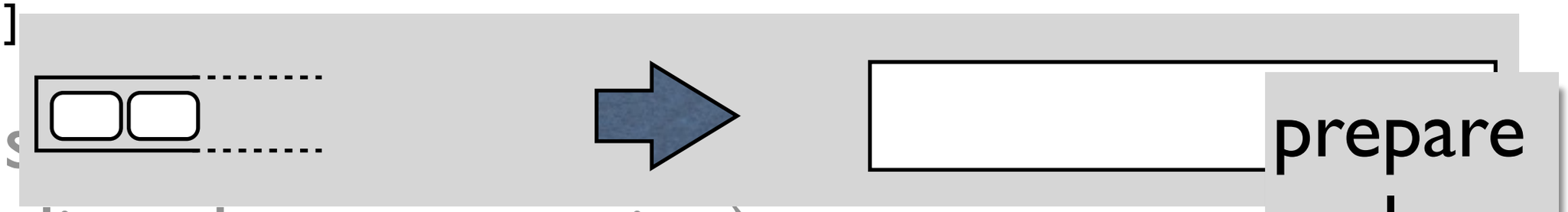
- purely incremental semantics contruction
  [(Peldszus et al. 2012, Peldszus & Schlangen 2012)]
- produces fully linked (*grounded in*) representations
- possible advantage: allows more interactions between (sub-)modules

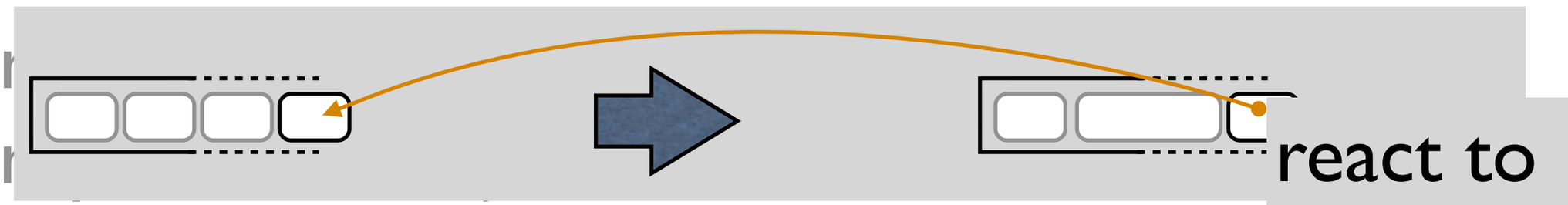


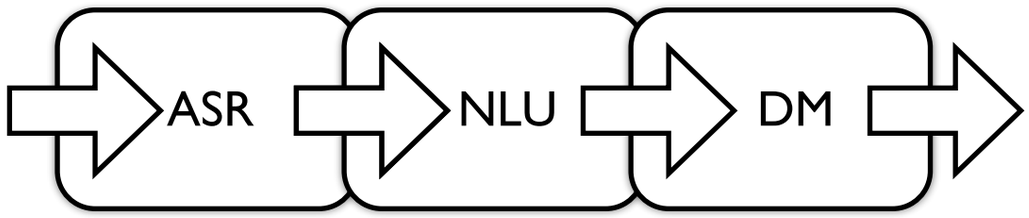
- if no interpretation found, try diff. parse

what can you do with it?

- predict whole representation: one (massively) multi-class problem
  [ICT (Sagae et al. 2009, DeVault et al. 2011, 2013), (Heintze et al. 2010)]
- s
  aligned representation) [(Heintze et al. 2010)]
- r
  r
- purely incremental semantics contruction







Part II

Challenges and Approaches

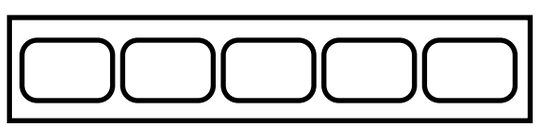
2.3 iDM

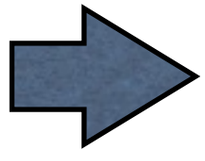
(incremental) DM

- input: semantic representation
- output: decision on system action
- the task: decide how to (re-)act
- incremental: input may not be based in complete utterance, may be revoked;
  within-turn actions possible

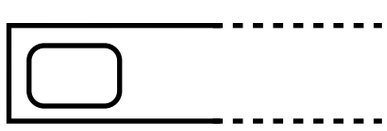
(incremental)

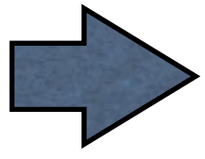
Dialogue Management



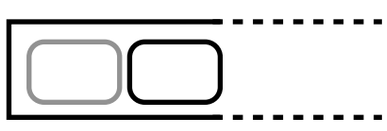


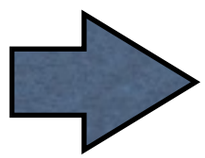
IS-Update,
Action Selection



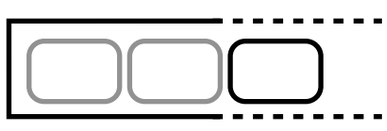


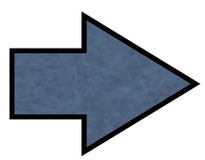
?



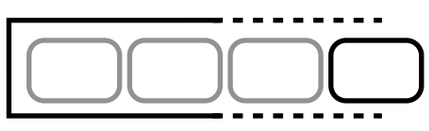


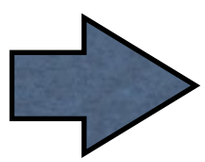
?



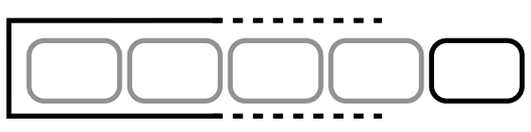


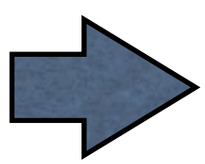
?





?

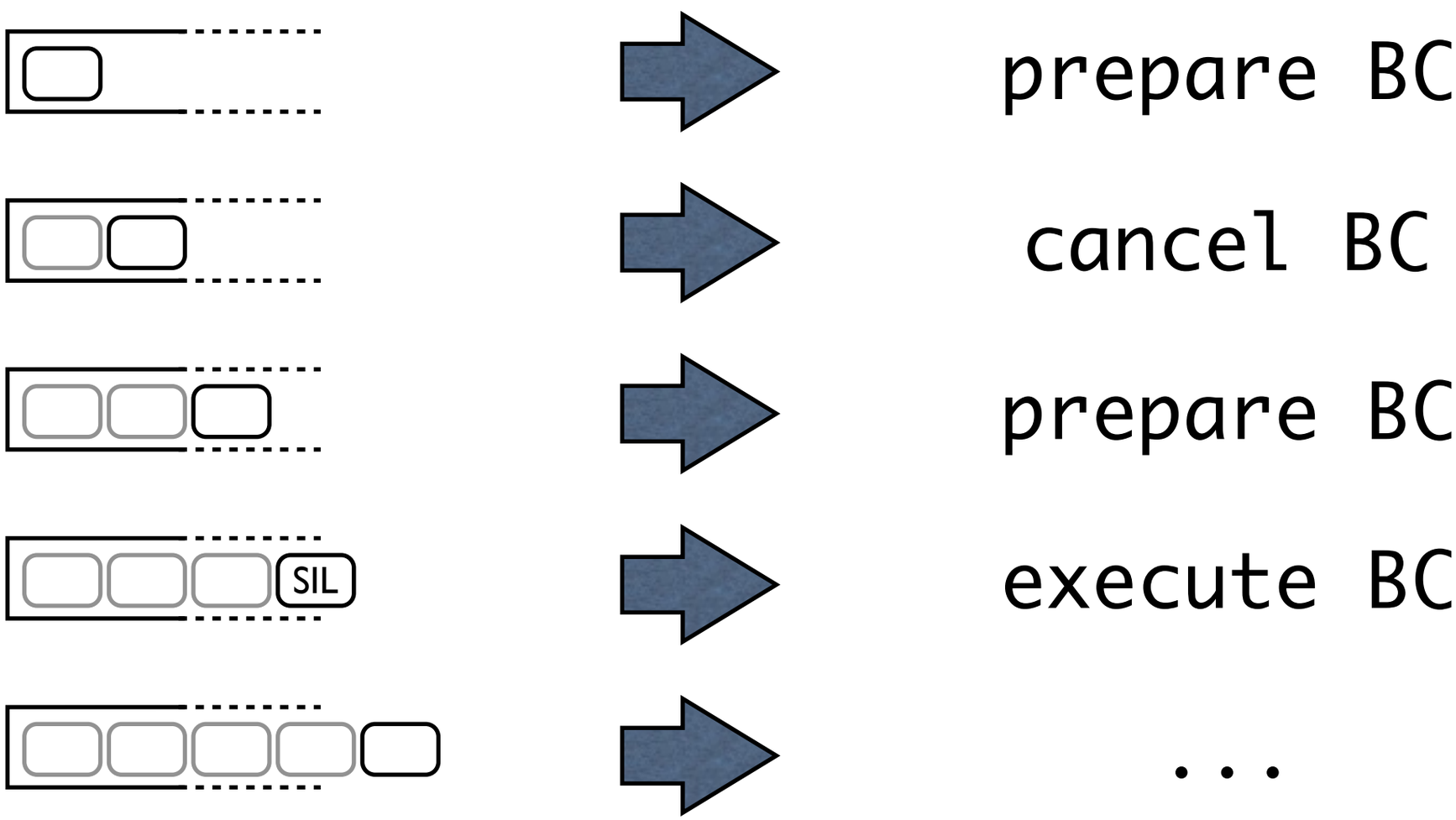




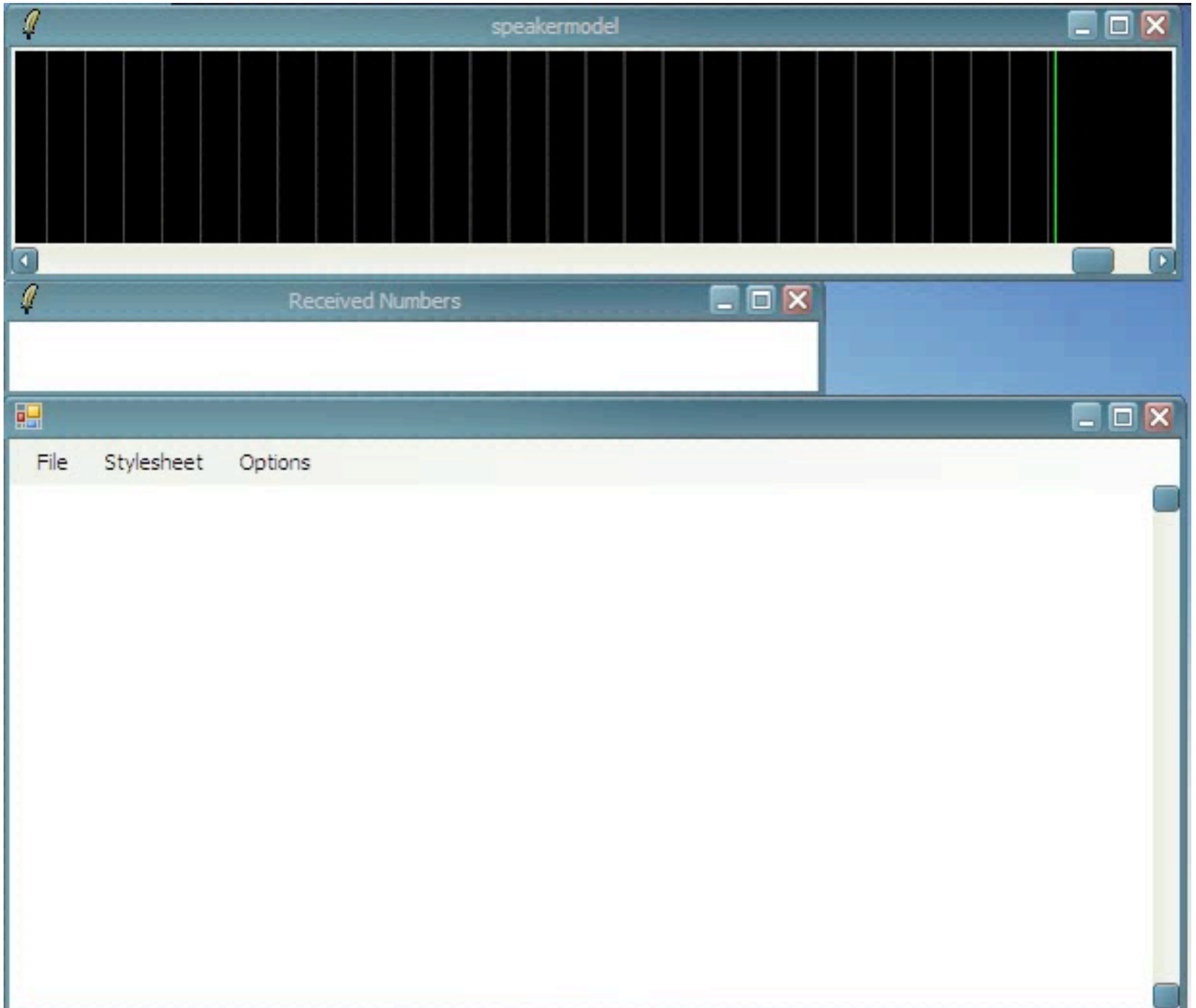
?

(incremental)
Dialogue Management

The *Numbers* system (Skantze & Schlangen, EACL 2009)

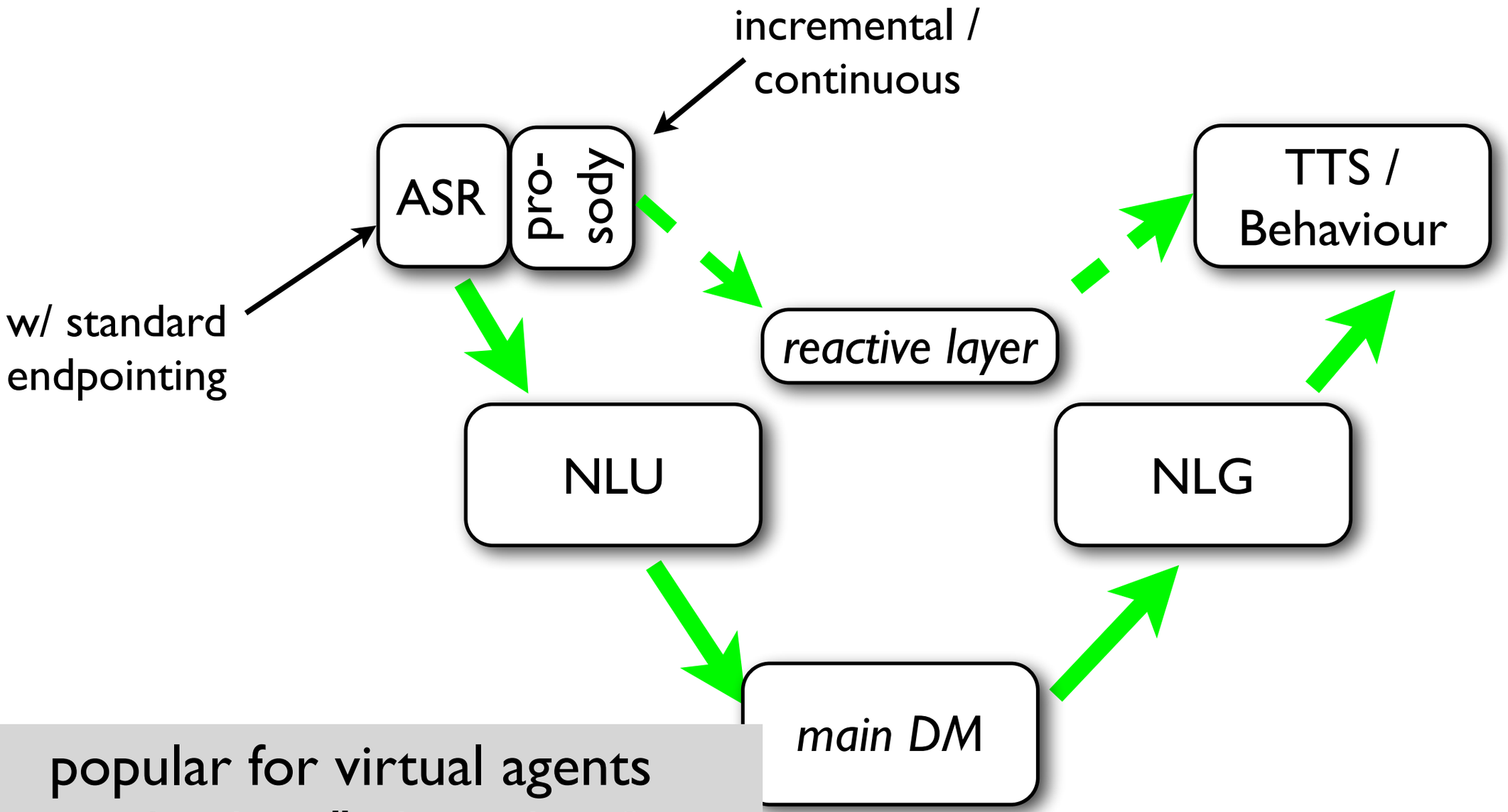


the *numbers* system



hybrid DM

separate incremental component, "normal" DM



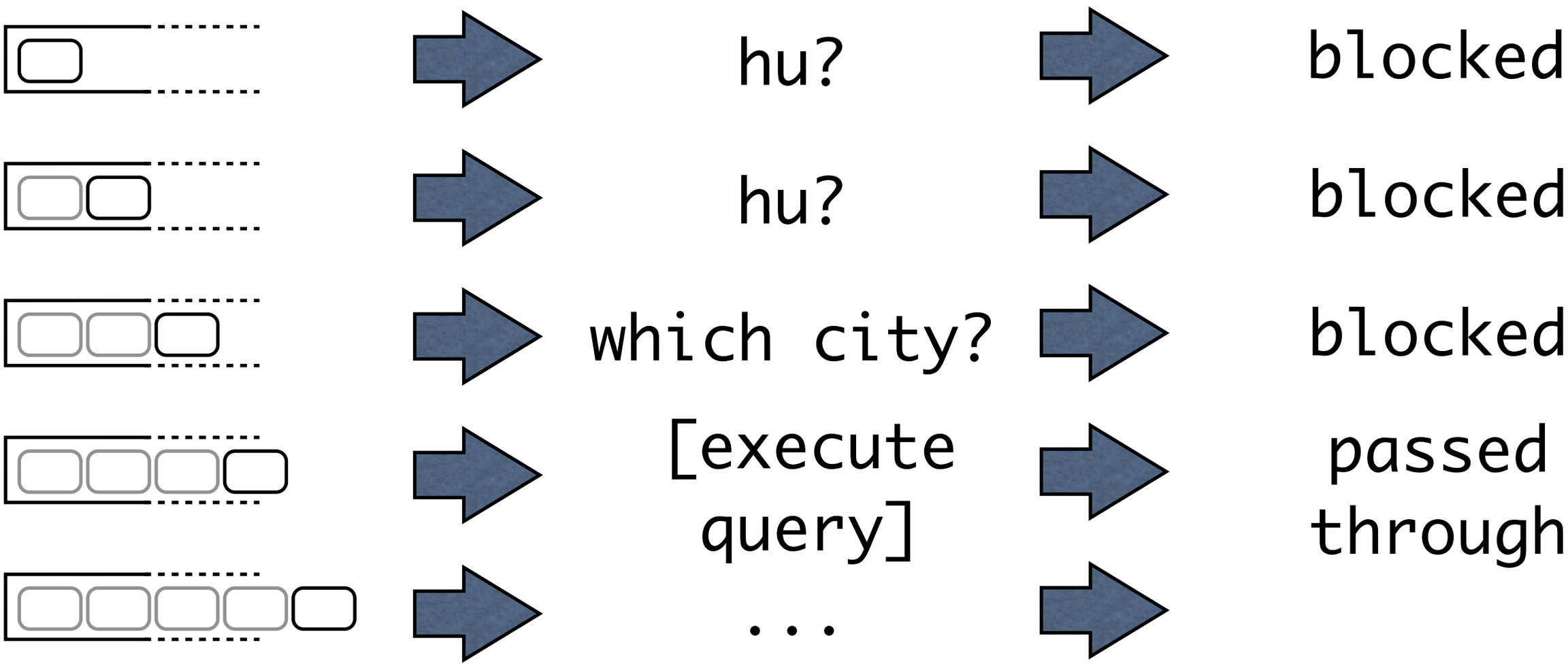
- popular for virtual agents
- can lead to "*mhm mhm Sorry, I did not understand..*"

(incremental)

Dialogue Management

Normal DM + Incremental Interaction Manager

(Selfridge *et al.* 2012; Khouzaimi *et al.* 2016)

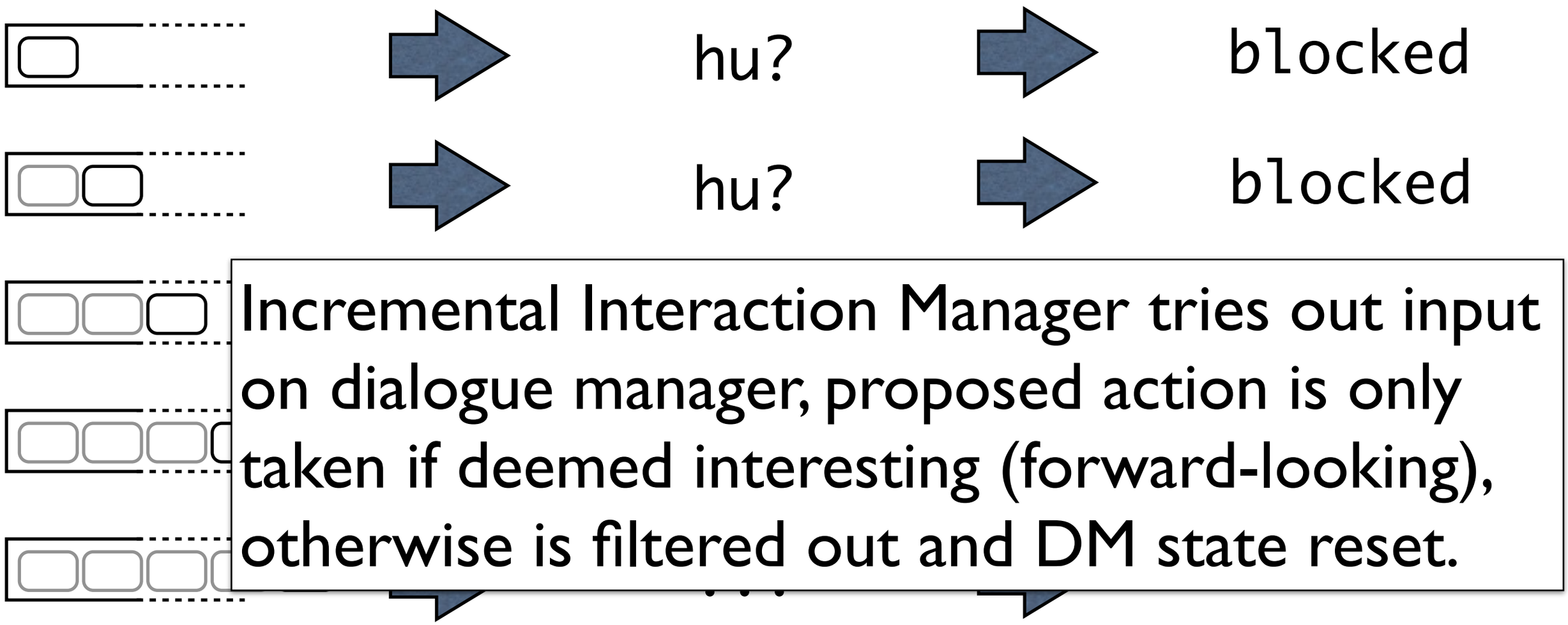


(incremental)

Dialogue Management

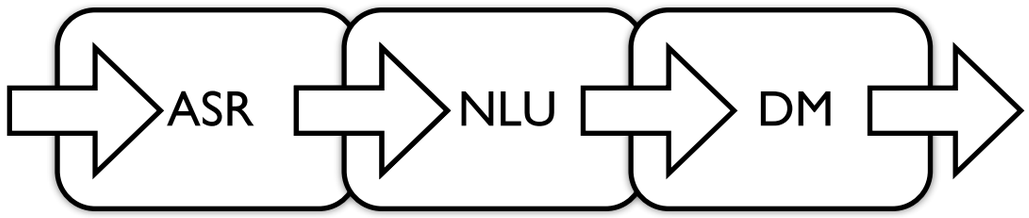
Normal DM + Incremental Interaction Manager

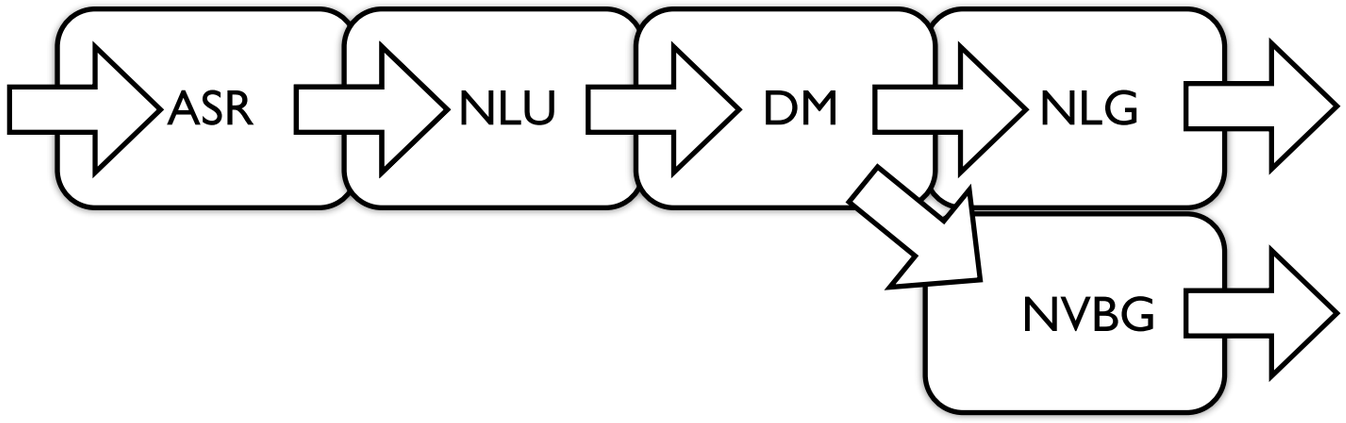
(Selfridge *et al.* 2012; Khouzaimi *et al.* 2016)



Summary DM

- incremental DM enables handling of additional behaviours (completions, delivery in installments)
- design space:
  - from keeping non-incremental DM, but adding more reactive second channel, to
    - real incrementality
- truly incremental DM decreases importance of notion of "utterance"; makes collaboration on utterances possible
- still an even wider open field, no standards yet, not really re-usable components
- (PO)MDPs??

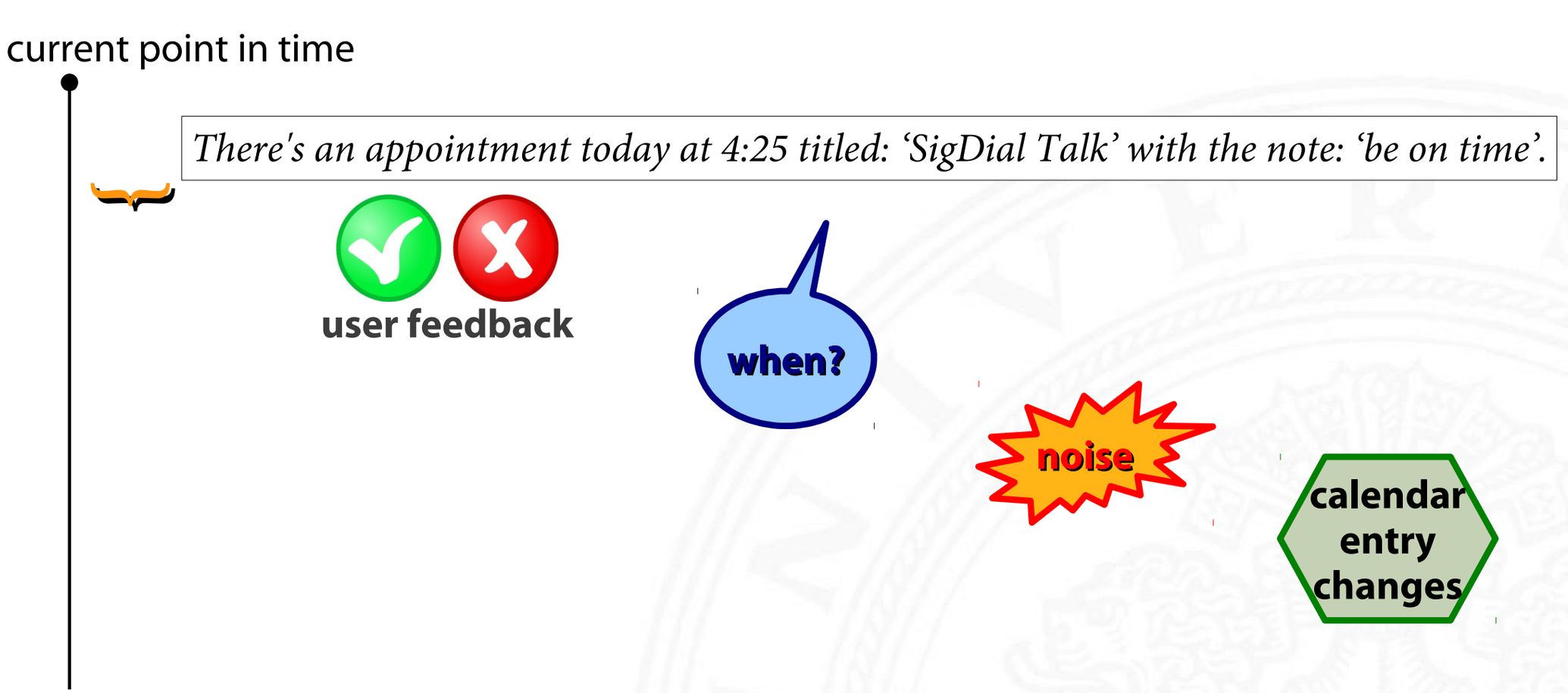

ASR
NLU
DM


ASR
NLU
DM
NLG
NVBG

Natural Language Generation (NLG)

Traditional approaches: all processing is utterance-initial

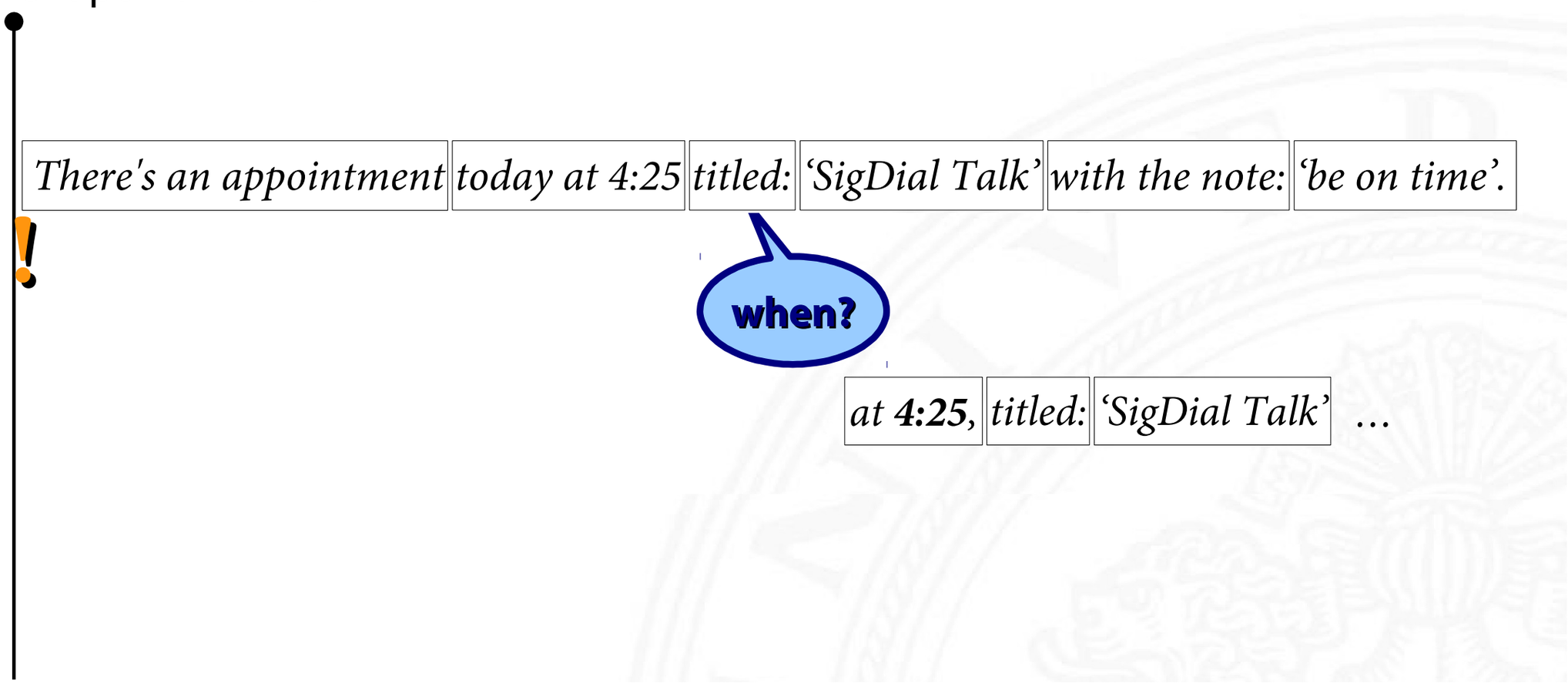
- potentially slow
- inflexible, unable to change to ongoing utterances



Incremental NLG

Potentially better to generate, synthesize and deliver in smaller *chunks*

- less utterance-initial processing — faster onset
- can take changes into account — react to feedback, requests, noise, …

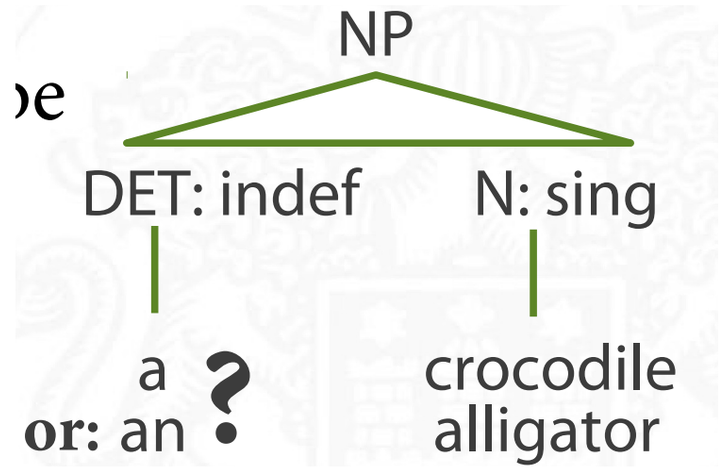


Incremental NLG

Granularity of chunks: size of incremental generation units?

- determines responsiveness to changes
- determines context available for further processing

- Smaller units?
  - ideally: word-by-word
  - but surface structure cannot be generated strictly left-to-right and word-by-word



- Bigger units?
  - enable coherent prosodic realization
  - fewer inputs lead to lower overhead
  - but limited responsiveness

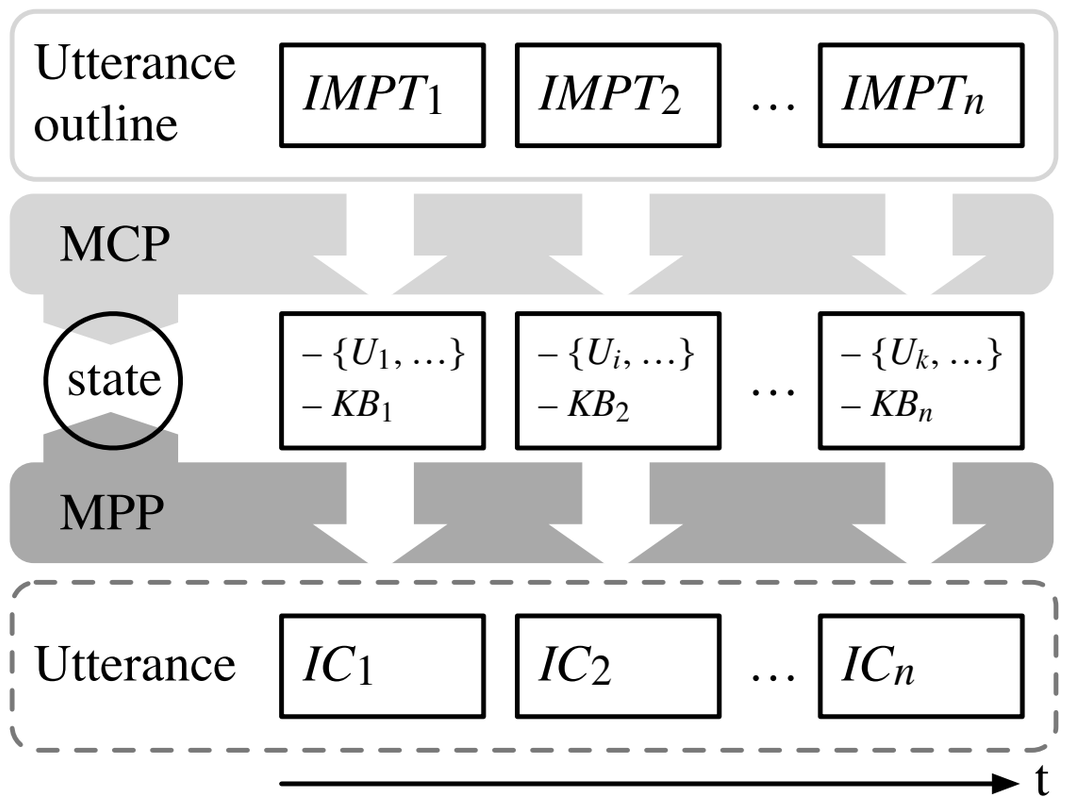
Incremental NLG

➔ sub-utterance chunk size

- corresponding to intonation phrases (roughly)
- *mildly* incremental generation

Approach: two stage planning process

- **micro-content-planning**: generates micro-planning tasks, chooses which one to generate next
- **micro-planning proper**: generates surface form for each IMPT, changes generation parameters
- communicate via a shared information state



(Buschmeier et al., SigDIAL 2012)

uses JavaSPUD (DeVault 2008)

From incremental to responsive generation

Responsive generation

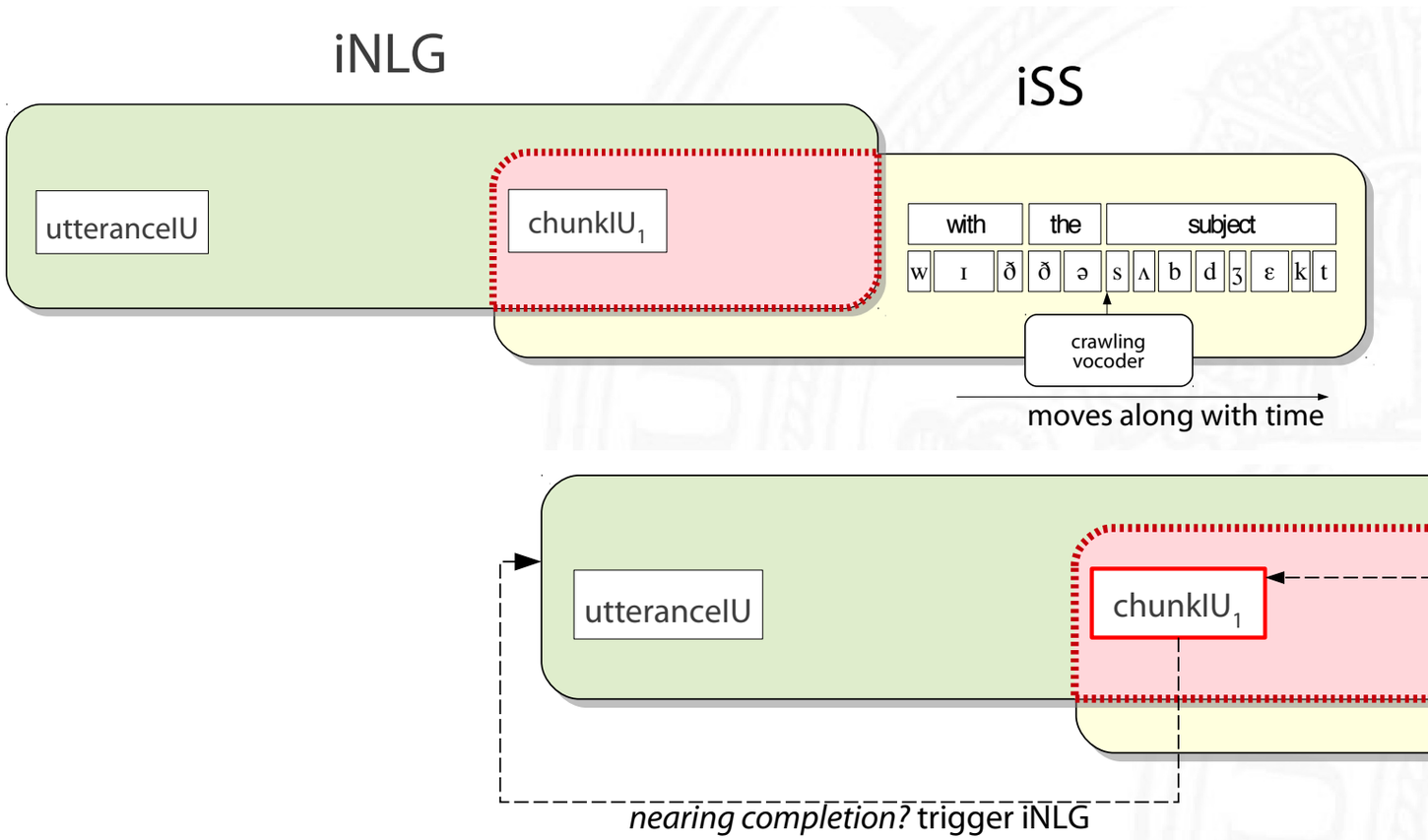
- incremental generation allows for dynamic, adapted creation of later sub-utterance chunks
- decisions about adaptations are delayed almost until the preceding increment finishes
- adaptation to state in both components
  - MCP: which IMPT next? repair/comment?
  - MPP: influence generation parameters, such as verbosity, redundancy

Example: verbosity

- length of utterance increment
- MPP uses predefined resources for desired degree of verbosity

iNLG + Speech Synthesis

- use of incremental speech synthesis (*INPRO_iSS*; Timo Baumann's course)
  - synthesizes just-in-time, some look-ahead to keep buffers filled



iNLG + Speech Synthesis

Results with iNLG + iSS (Buschmeier et al., SigDIAL 2012):

- reduces latency over a non-incremental baseline

processing step	non-incr.	incremental
NLG	361	52
Synth. (ling. processing)	217	222
Synth. (HMM & vocoding)	1004	21
Total	1582	**295**

- information presentation of calendar entries, with random noise: adaptive presentation after noise is rated more natural
  stop-and-restart >* stop-and-wait ~ ignore-and-continue

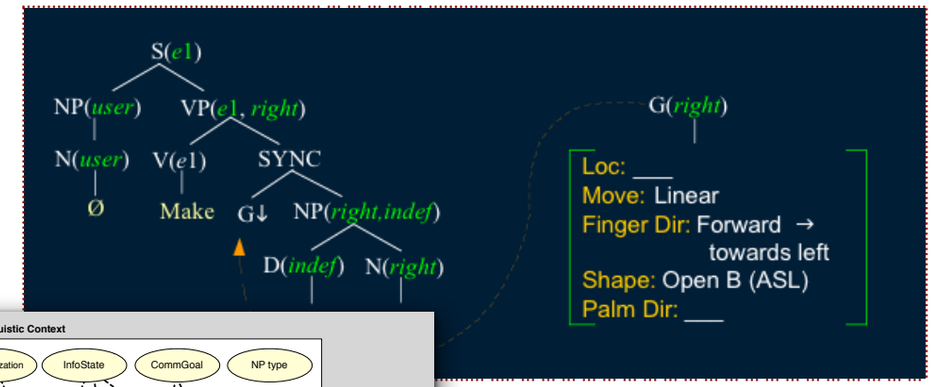
NVBG — nonverbal behavior generation

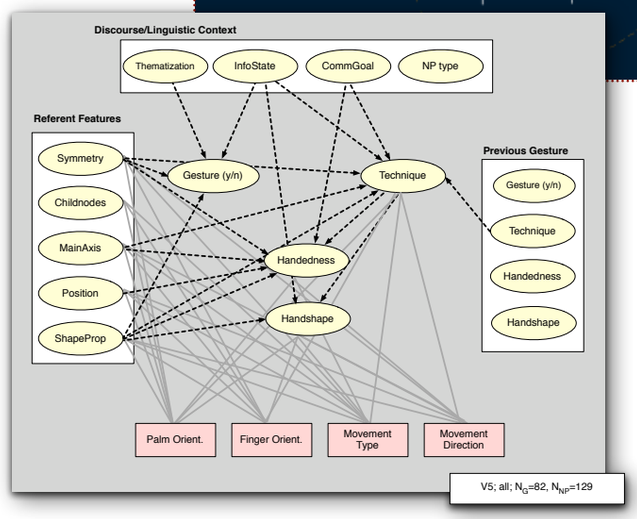
- **Task**: Generation of nonverbal behaviors
  - selection, coordination („fission“), synchronization
  - as a function of intended meaning, dialogue function, discourse function, speaker state, information state, …

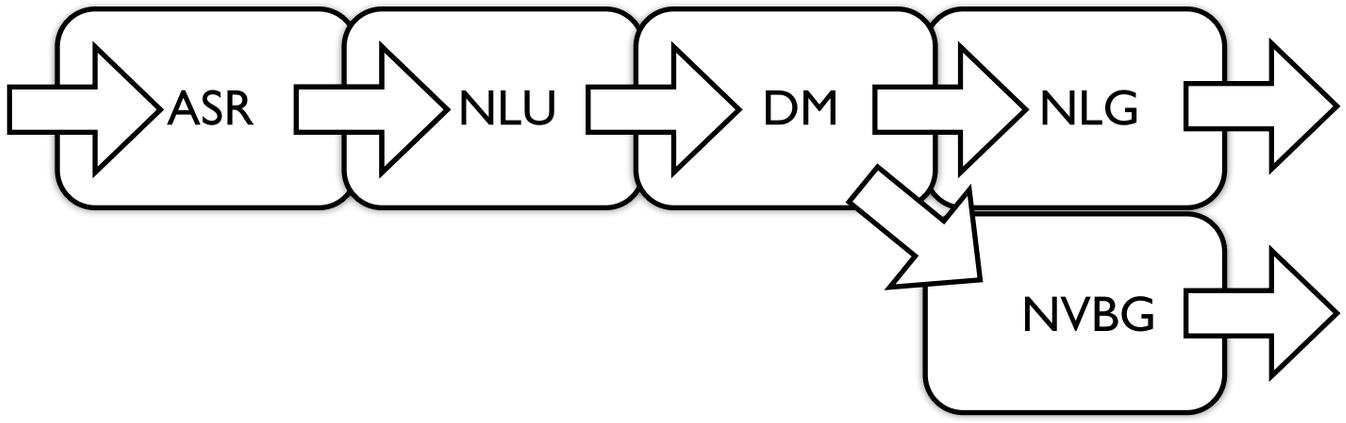
- Early approaches (Cassell et al. 2001) and current practical ones use simple formalism (rules or transducers; Marsella et al.) to formulate mapping

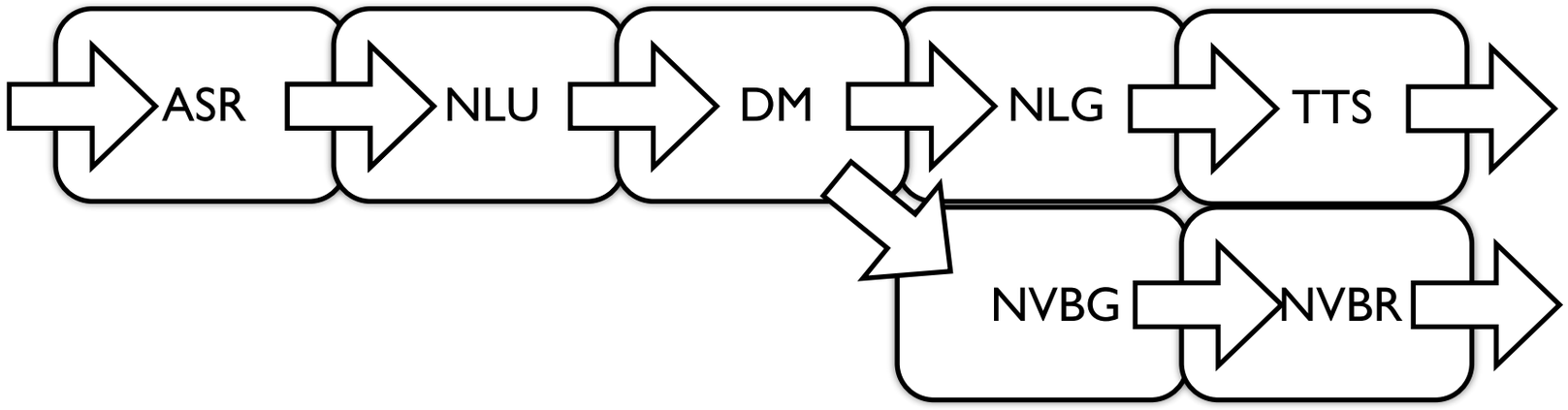
- Integrated microplanning (multimodal grammar) (Kopp et al. 2004)

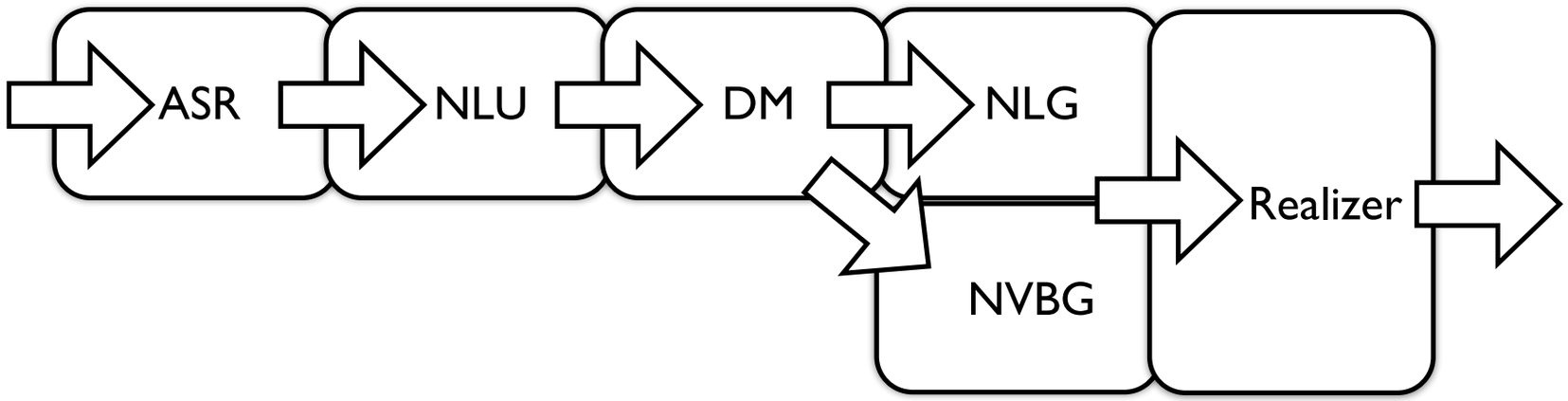
- Recent approaches focus on one or few modalities, learned from data






ASR
NLU
DM
NLG
NVBG


ASR
NLU
DM
NLG
TTS
NVBG
NVBR

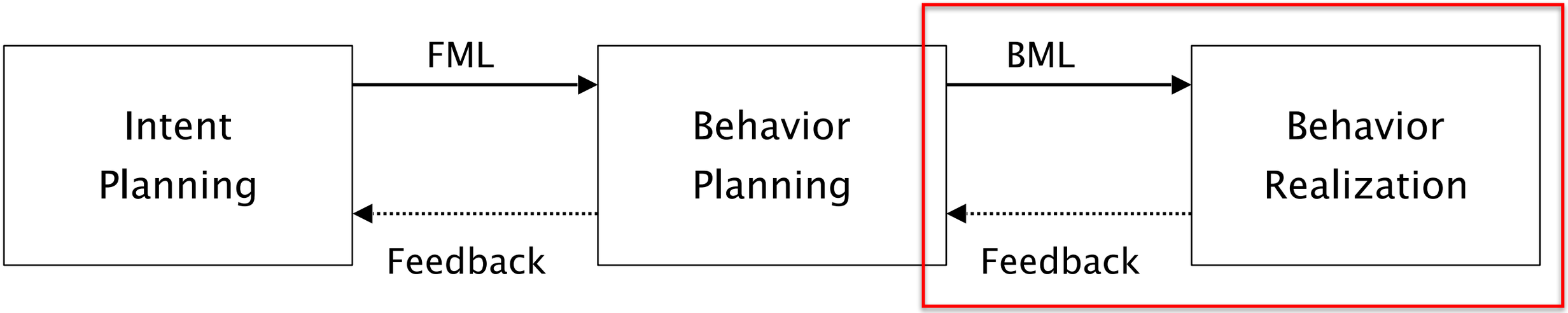

ASR
NLU
DM
NLG
Realizer
NVBG

Incremental behavior realization

- Final task in an end-to-end system: realize behavior into perceivable output
  - speech, prosody — text-to-speech synthesis
  - nonverbal behavior (face, gesture, gaze, head, posture, …) — computer graphics for virtual agents, motor control for physical robot agents
  - other modalities/media — visualization, acoustic cues, …

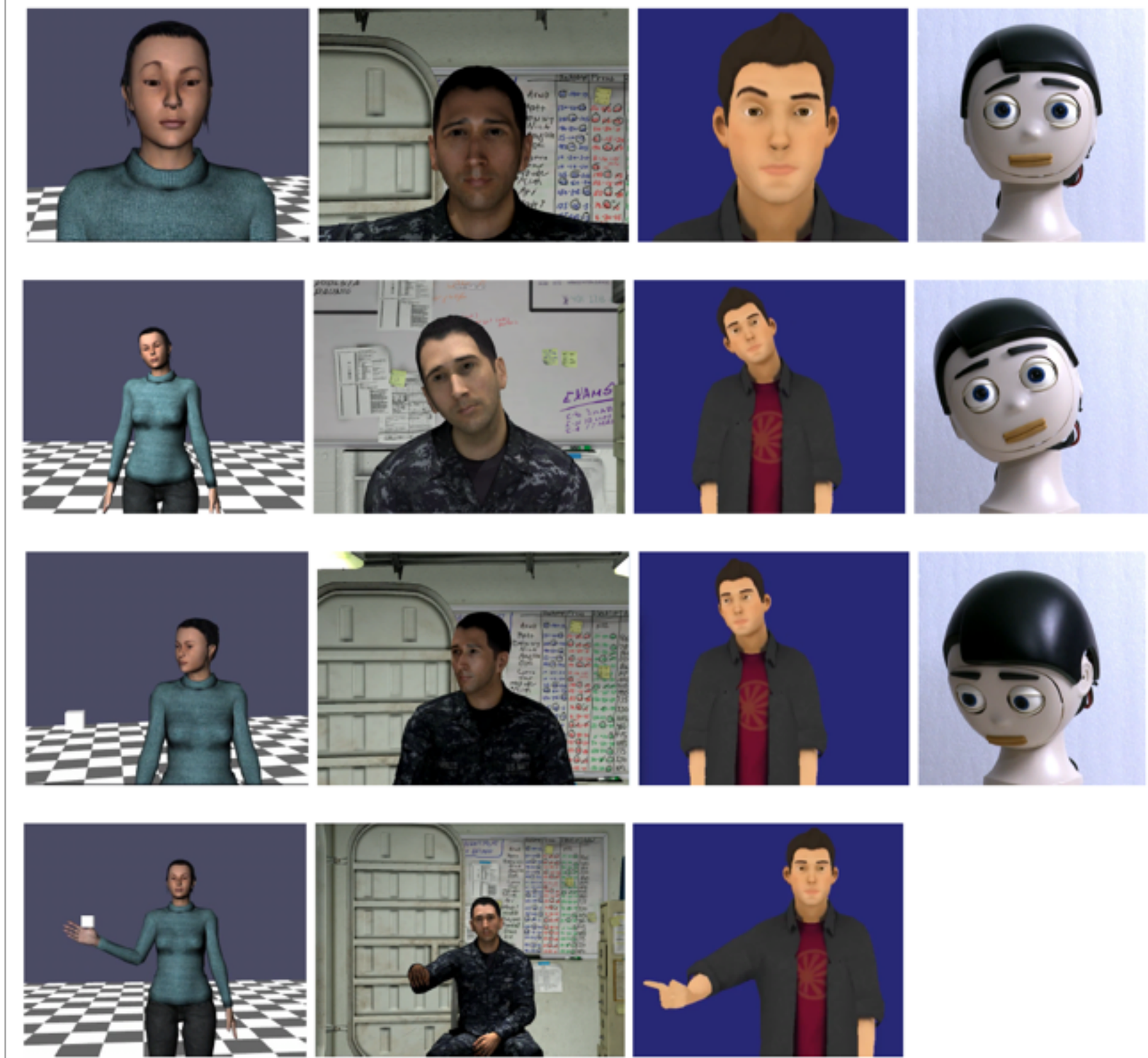
- Main challenges, often in trade-offs
  - **quality**: expressivity, intelligibility, naturalness, lifelikeness, sample rate, …
  - **efficiency**: latency, computational cost (time, memory)
  - **flexibility**: controllability, adaptivity to external or internal constraints, …
  - **synchrony**: internal coherence (e.g. temporal coordination) between modalities, sync with external events

SAIBA framework



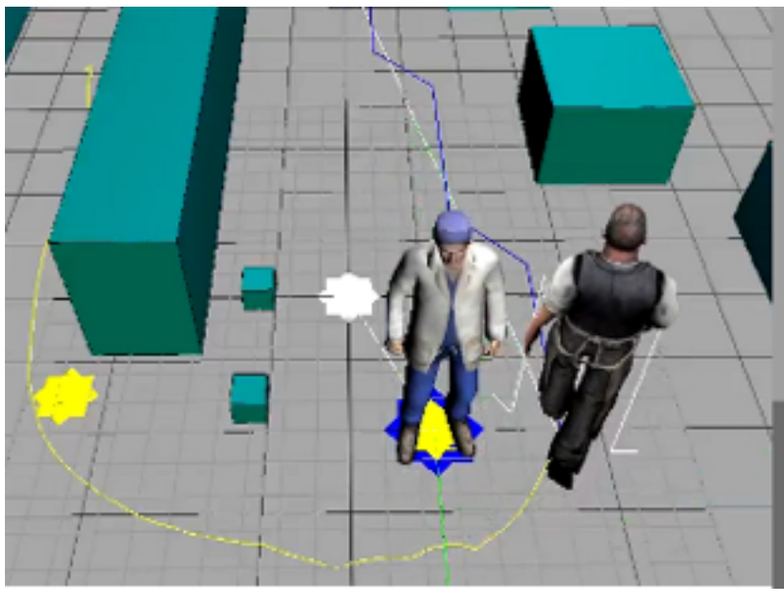
- Three-stage structure of behavior generation in many existing ECAs
- Idea: modularization and separation of stages (treated as black boxes)
- Enable interoperability and exchange of modules
- Definition of interfaces between stages — common markup languages
  - **Behavior Markup Language (BML)**
  - Function Markup Language

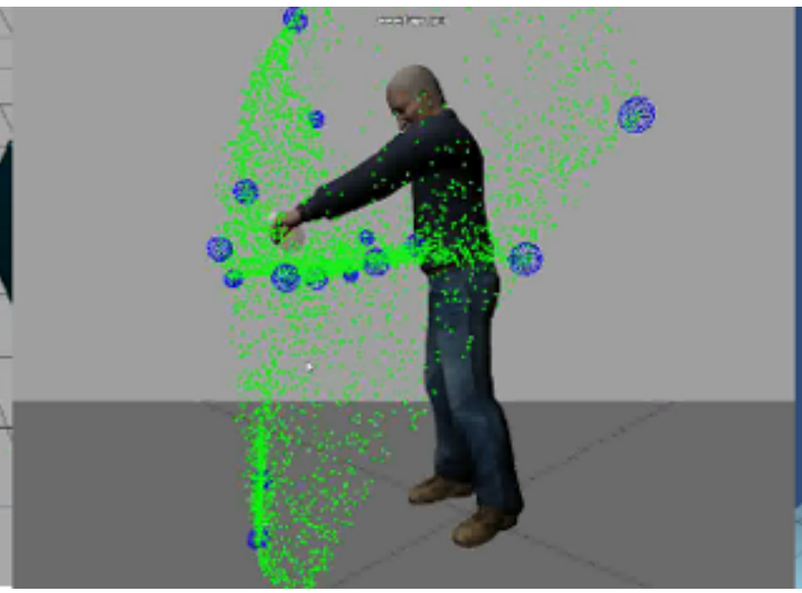
Different realizers, one BML



Smartbody (ICT, USC LA)

- http://www.smartbody-anim.org/
- Focus: very realistic behavior
  - Motion Capture or artist created animations
  - Support for recorded voices



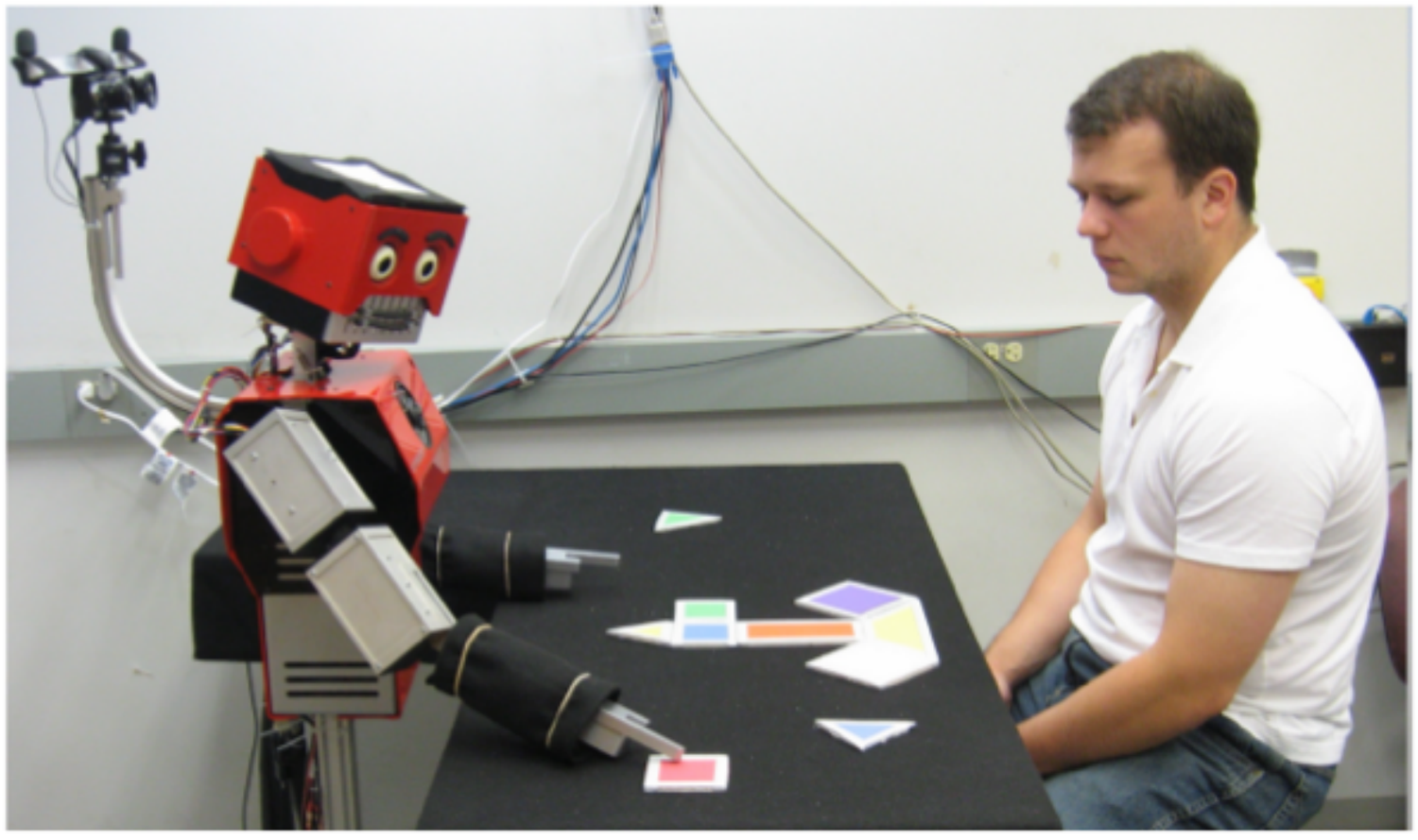


LiteBody

- http://relationalagents.com/litebody.html
- Webbased, 2D,lightweight
- Used in long-term studies
- Robust

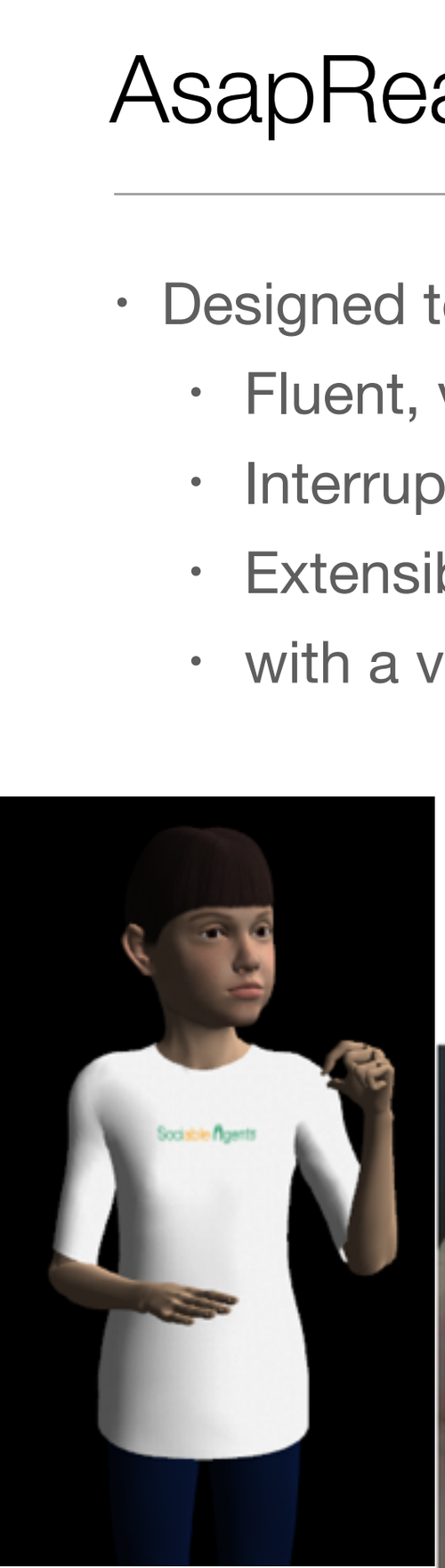
ROS BML Realizer

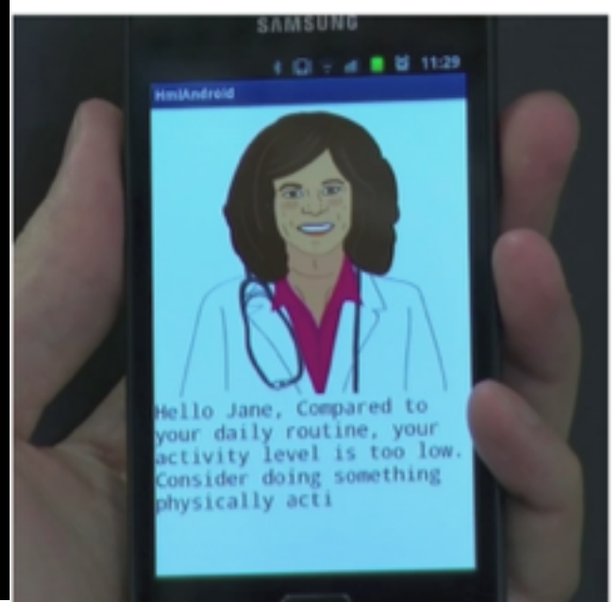
- http://sourceforge.net/projects/rosbmlrealizer/
- Uses the Robotic Operation System (ROS)
- realizes BML on robot body



AsapRealizer

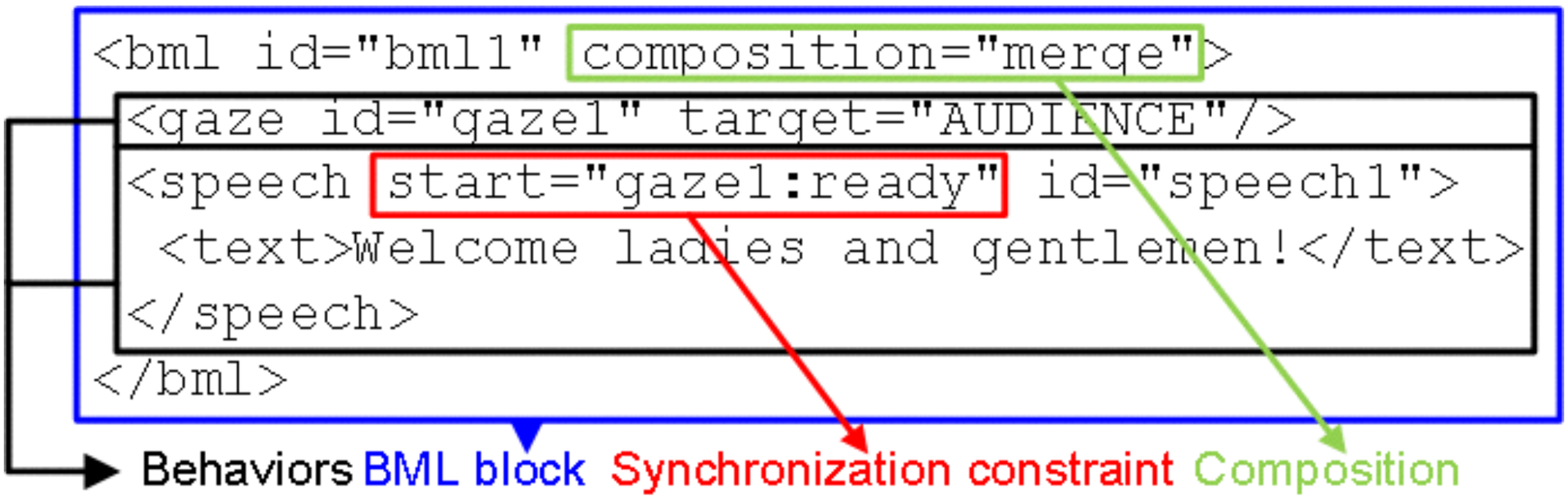
- Designed to allow fluent interaction
  - Fluent, very interactive behavior realization
  - Interruptions, on-the-fly-adaptation, incrementality, reactivity
  - Extensibility
  - with a virtual human or robot





BML design

- Describes occurrence of behaviors
- Relative timing of behaviors
- Form of behaviors
- Realizer-independent
- But allows extensions for realizer-dependent behavior



BML example

- Specification of a co-speech deictic gesture

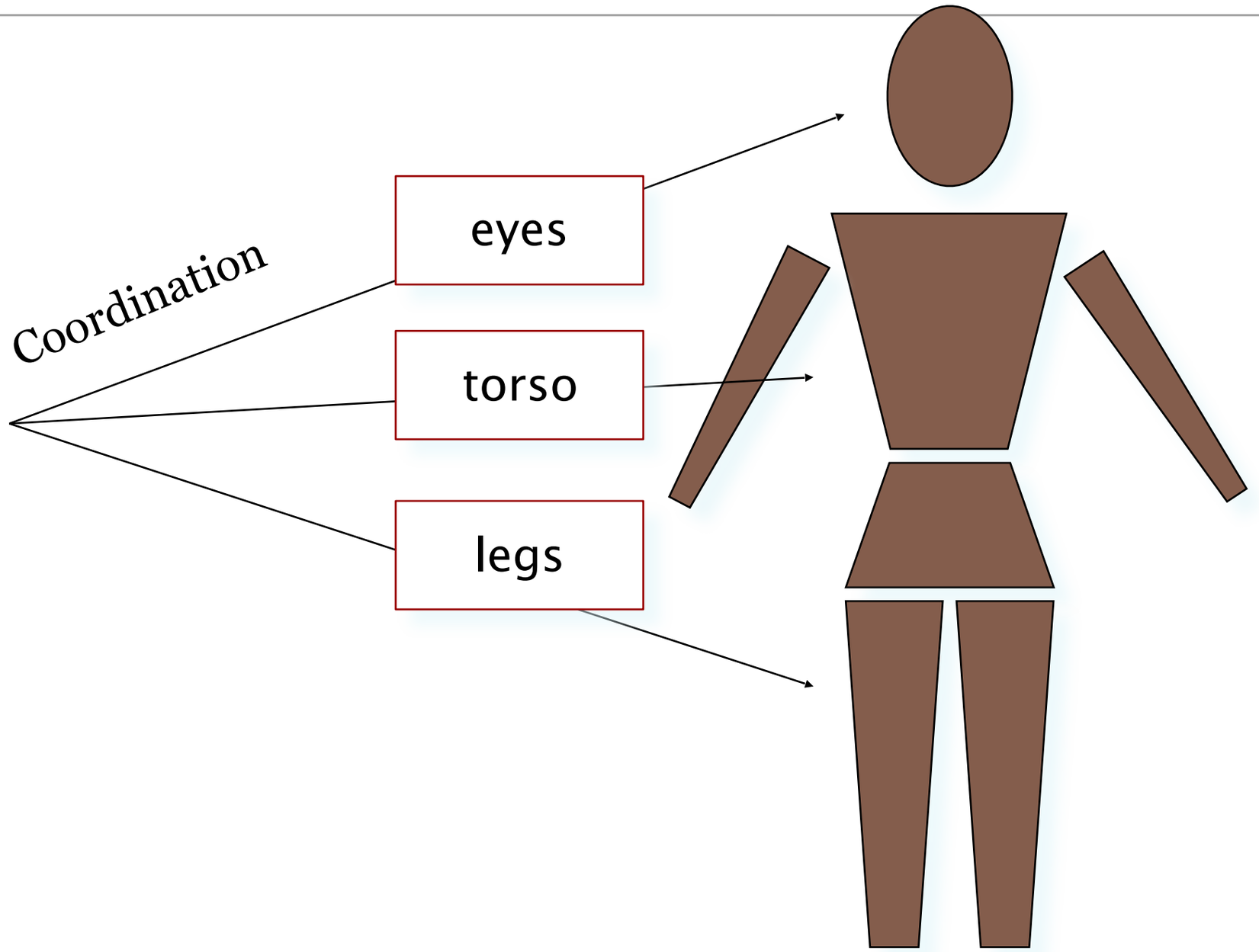
```
<bml
  xmlns="http://www.bml-initiative.org/bml/bml-1.0"
  id="bml1">

  <speech id="speech-1">
    <text>
      Look, it's over <sync id="sync-1" /> there.
    </text>
  </speech>

  <pointing
    id="point1"
    ready="speech-1:sync-1"
    mode="LEFT_HAND"
    target="camera" />

</bml>
```

behaviours

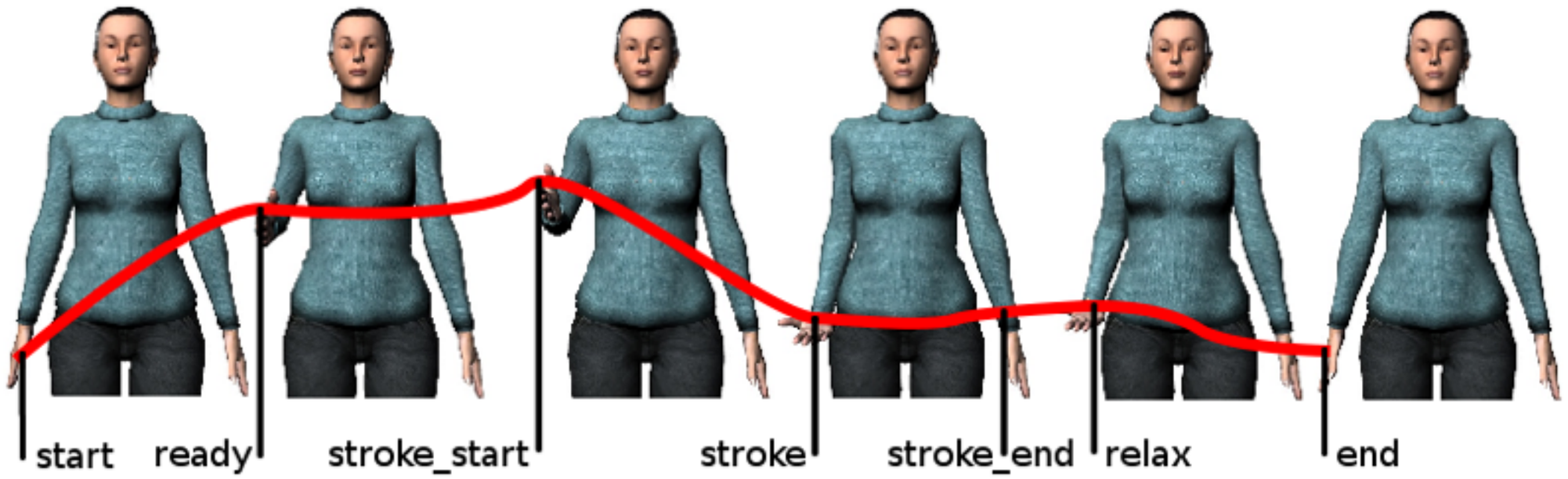
syncronisation constraint

BML behaviors

- Gesture
- Head
- Gaze
- Speech
- Locomotion
- Posture
- Facial expression



BML phases and sync-points




```
<bml>
   <gaze id="gaze1" target="AUDIENCE"/>
   <speech start=”gaze1:ready” id="speech1">
       <text>Welcome ladies and gentlemen!
       </text>
   </speech>
</bml>
```

BML feedback from realizer

- To provide the behavior planner with information on
  - Delivered behaviors: **Progress** feedback
  - Delivery failures: **Warning** feedback
  - Predicted timing and form decisions: **Prediction** feedback

To realizer:

```
<bml id="bml1">
 <gesture id="b1" lexeme="BEAT"/>
</bml>
```

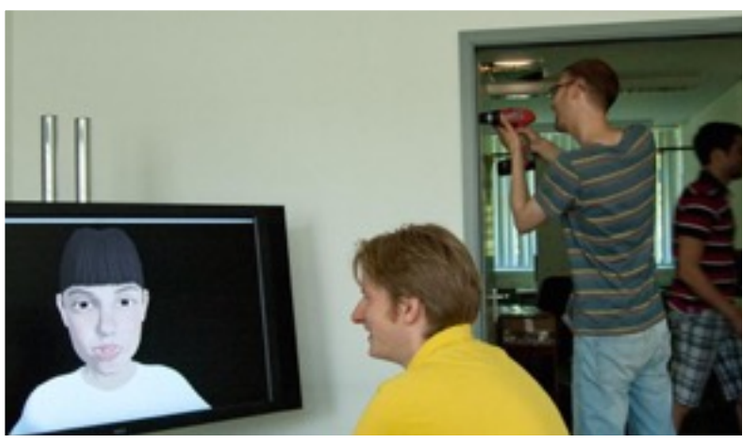
From realizer:

```
<predictionfeedback>
 <gesture id="b1" lexeme="beat" mode="RIGHT_HAND"
          start="0" ready="1"
          strokeStart="1" strokeEnd="2"
          relax="2" end="3"/>
</predictionfeedback>
```

Behavior realization for responsive agents

We require the realizer to enable a lot of things:

- Mid-utterance (self-)interruption
- Seamless turn-taking (i.e. respond quickly to external events)
- Fighting over the turn using louder speech, speeding up/slowing down, …
- Responding to listener feedback, e.g. delaying speech until the listener has finished speaking or resuming before their delivery is finished
- Employing fillers to keep or attain the turn, without having a full plan at hand
- Retain multimodal synchrony when adapting a behavior
- …

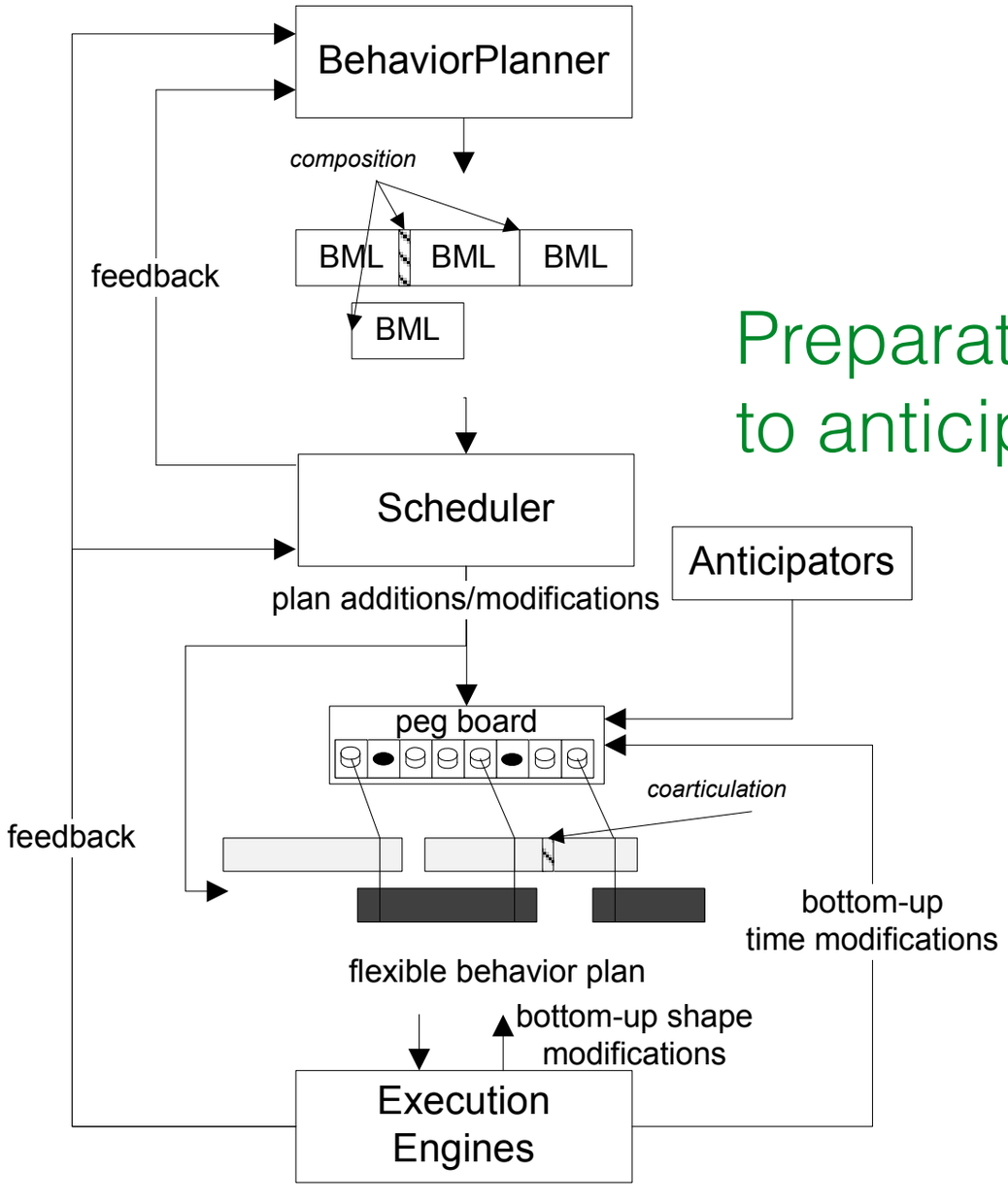


Incremental behavior realization: ASAP

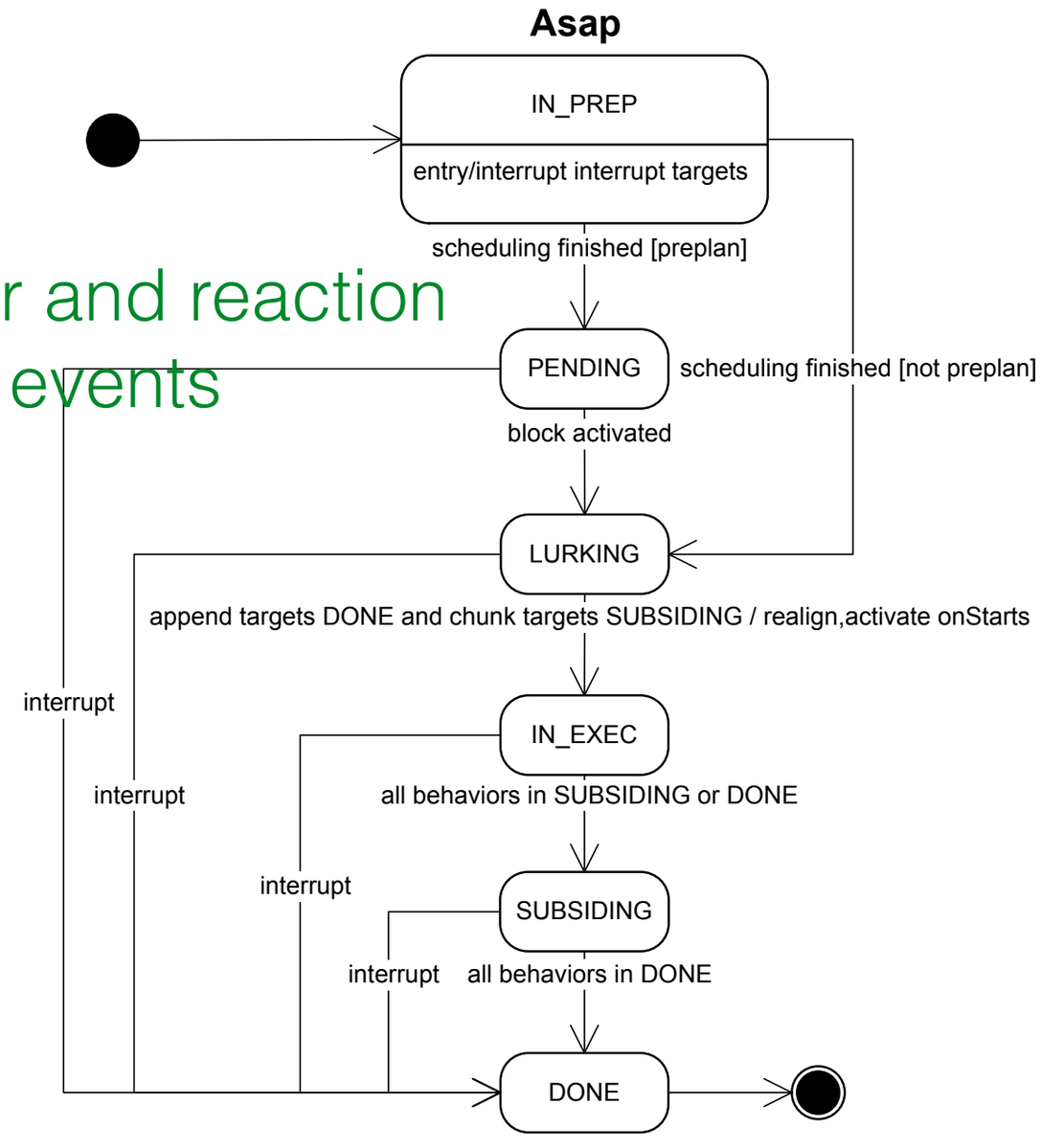
- BML extensions BMLA and BMLIS to enable incremental, adaptive and interruptive speech and behavior realization

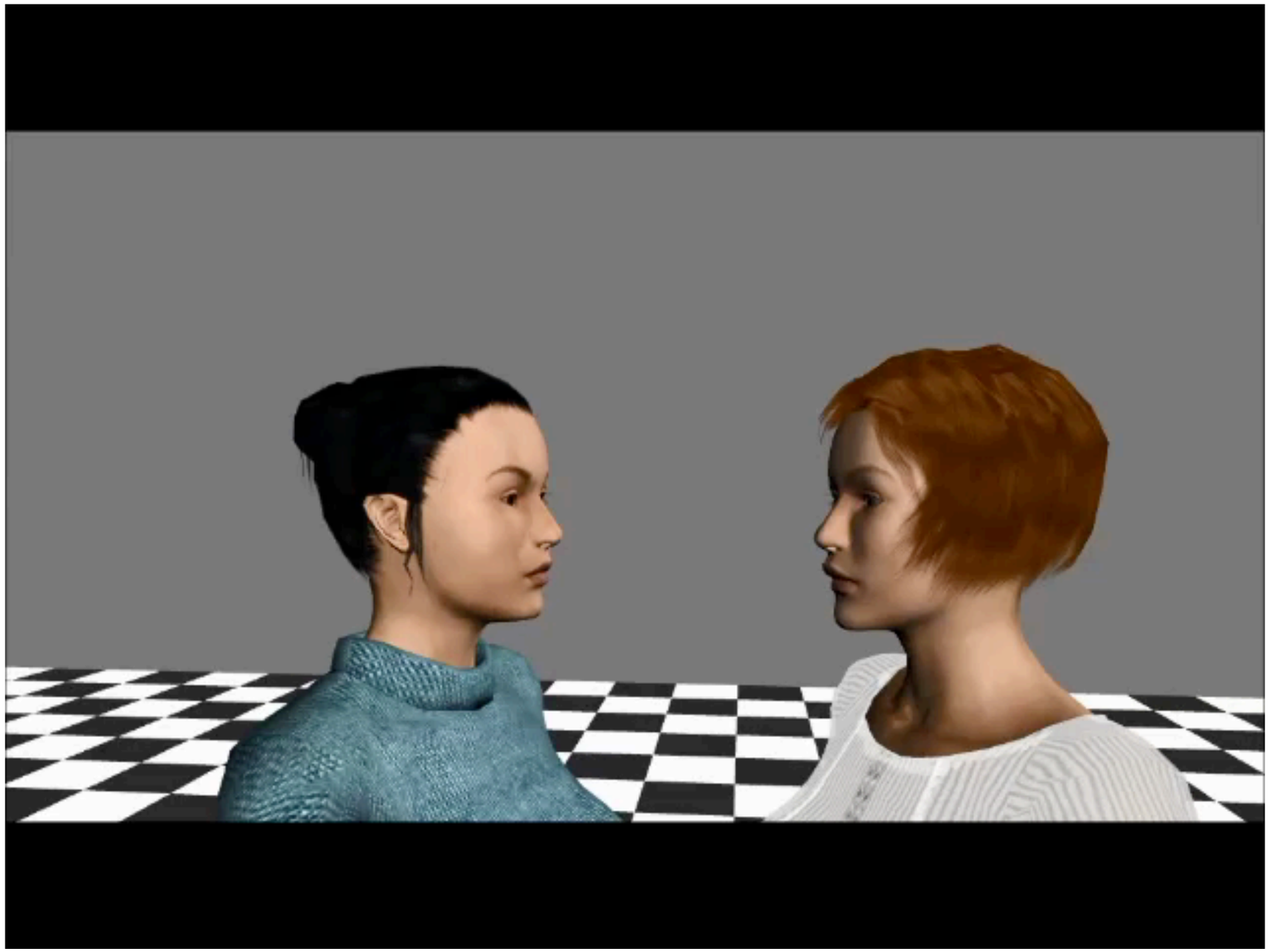
- **ASAP realizer** *(*artificial social agents platform*)*
  - incremental construction of plans
  - continuous modification of the timing and shape of ongoing behavior
  - fluent connection of increments
  - interface to Inpro_iSS and other TTS engines, animation engines, robots

ASAP realizer architecture



Preparation for and reaction to anticipated events





Example: Modeling turn taking dynamics

Extensions for fluent interaction

- **Interruption** — *more than just stopping*
  - find earliest feasible interruption points
  - gracefully remove behavior

```
<bmla:interrupt id="i1"
  target="bml1"
  start="shake1:stroke"
  exclude="speech1,gesture1"/>
```

- Parameter **value change**
  - even at execution time
  - for running behavior

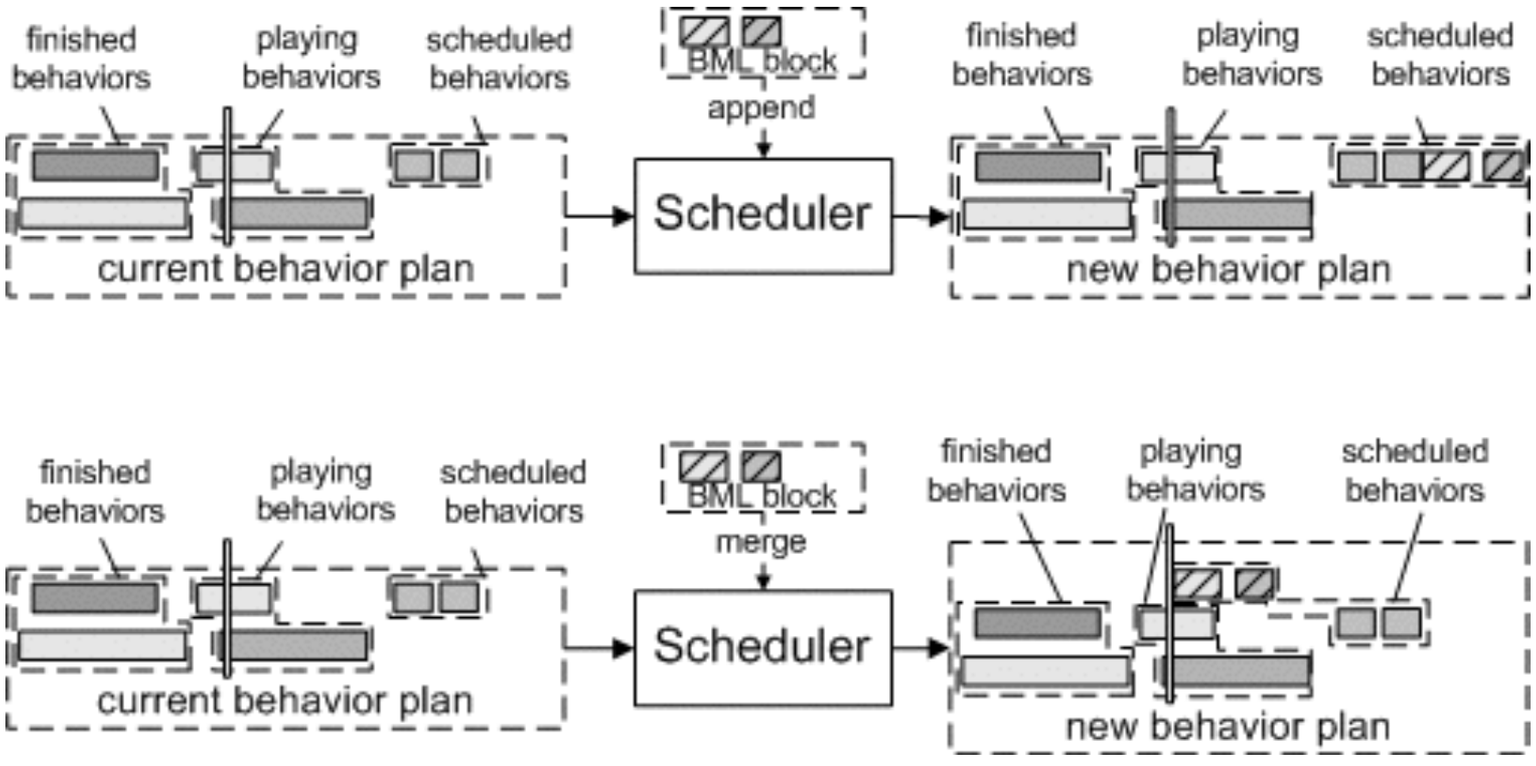
```
<bmla:parametervaluechange id="p1"
target="bml1:speech1" paramId="volume"
start="bml1:speech1:sync1"
end=„bml1:speech1:sync1+1"/>
```

Extensions for fluent interaction

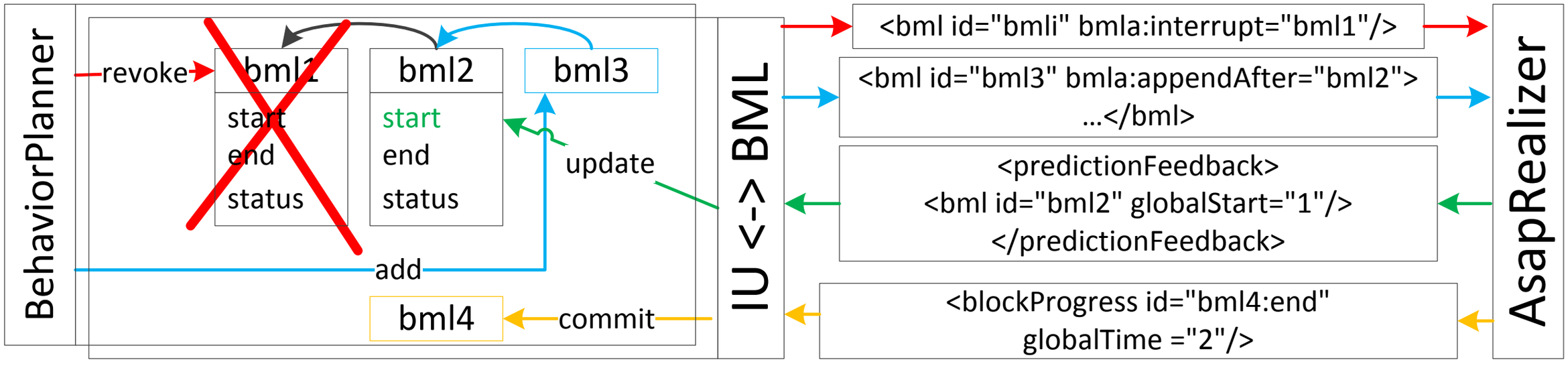
- **Incremental composition**
  - compose behavior out of smaller BML blocks
  - fine-grained composition: append/prepend, chunk before/after

```
<bml id="bml3"
bmla:appendAfter="bml1,bml2"
bmla:prependBefore=„bml4"/>
```

```
<bml id="bml3"
bmla:chunkAfter=„bml1,bml2"/>
```



Example: incremental planning and realization



Overview of Day 2

- Dialogue Processing Flow: ASR —NLU — DM — NLG / NVBG — Realizer
  - all components must run incrementally and interact via local updates
- IU model:
  - IS updated with minimal units of information, as soon as hypothesised
  - “Higher-level” hypotheses formed on basis of “lower-level” ones
  - IS may have to be revised, in light of newer information
- Hybrid system / DM: *main* DM + reactive layer
- Incremental generation is faster and adapts more naturally to disturbances
- Incremental realization requires plan construction, interruption, continuous modification, fluent connection of increments, based on prediction of events

Questions?

End of Day 2

Tomorrow: Introduction to technical framework

Literature

- http://www.dsg-bielefeld.de
- http://scs.techfak.uni-bielefeld.de

- Branigan, H. P., Catchpole, C. M., & Pickering, M. J. (2011). What makes dialogues easy to understand? Language and Cognitive Processes, 26(10), 1667–1686. doi:10.1080/01690965.2010.524765

- Clark, H. H. (1996). Using Language. Cambridge, England: Cambridge University Press.