UiO : University of Oslo

Dialogue management for rich, open-ended domains: *a few preliminary ideas*

Pierre Lison
University of Oslo, Dep. of Informatics

Språkteknologisk seminar @ LNS
January 24, 2011

Introduction

- My primary objective for this talk is to present you my ongoing PhD work
  - Focus: *dialogue management* for complex domains
  - Of course, most of the ideas I am going to present today are still in their infancy
  - Feedback (during & after the talk) *most* welcome!
- I'll also tell you a few things about the research I have been doing before coming here
- *Feel free* to interrupt me for questions or comments

Outline of the talk

- Background
  - Generalities about dialogue systems
  - Dialogue management, dialogue models
  - Learning dialogue policies
- Ongoing work
  - General vision
  - Current research directions
  - Implementation & evaluation
- Conclusion

Outline of the talk

- **Background**
  - **Generalities about dialogue systems**
  - Dialogue management, dialogue models
  - Learning dialogue policies
- Ongoing work
  - General vision
  - Current research directions
  - Implementation & evaluation
- Conclusion

Generalities about dialogue systems

- A (spoken) dialogue system is a computer-based system able to interact with human users using (spoken) natural language
- Prototypical "intuitive/natural" interface: the only thing you need to do to use a dialogue system is to *speak*
- Expected to play an ever-increasing role in our daily interactions with technology
  - telephone-based systems for information and service
  - language-based tutoring systems
  - speech-enabled software applications
  - service robots in homes, schools, offices or hospitals
  - etc.

[Zue,V. 1997]

[Goodrich, M.A. and Schultz, A.C. 2007]

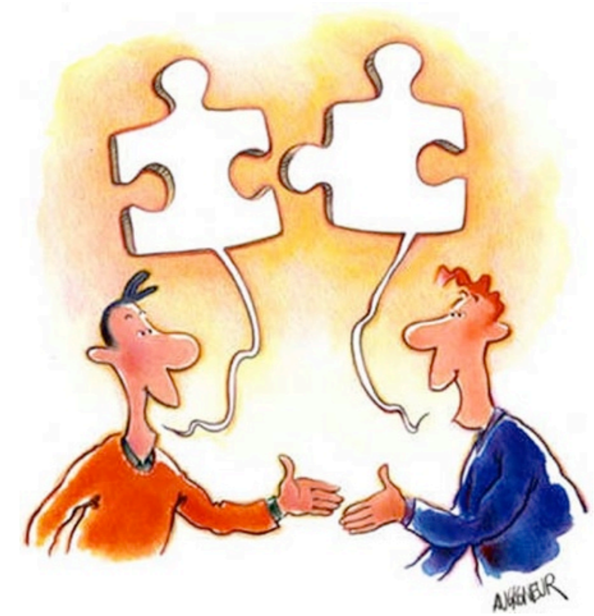
Generalities about dialogue systems

- Dialogue is a natural medium of communication, but it is also a quite complex one to process!
- Unsurprisingly, developing a spoken dialogue system (or SDS for short) can therefore be a demanding enterprise
- SDS generally depend on the availability of a wide range of natural language technologies:
  - automatic speech recognition (ASR);
  - syntactic parsing & semantic interpretation;
  - dialogue management;
  - natural language generation;
  - speech synthesis

[Jokinen, K. & McTear M. 2009]

What is dialogue, anyway?

- What is **dialogue**?
  - Spoken ("verbal") and, possibly, non-verbal interaction between two or more participants
  - "Language as action": Dialogue is a social *activity*, serving one or several purposes for the participants (promising, ordering, warning, asking, keeping social contact, etc.)
  - Dialogue also expresses *meaning*, which needs to be understood by all participants in a successful interaction
  - This is realised via the gradual expansion & refinement of the *common ground* (shared background knowledge)

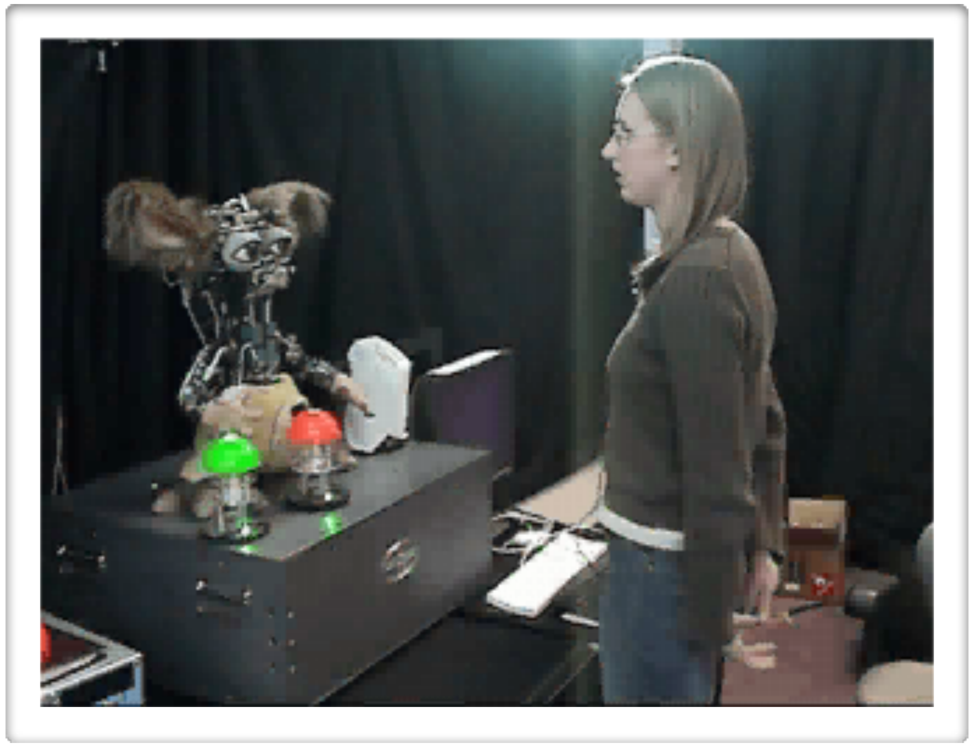


What is dialogue, anyway?

- Spoken dialogue is often *referential* to some (proximate or distant) spatio-temporal context
  - The context might be a small- or large-scale environment, a task to perform, a software application, etc.
  - *Grounding* dialogue thus requires the ability to resolve linguistic references to the (own) situation awareness and yield a common ground

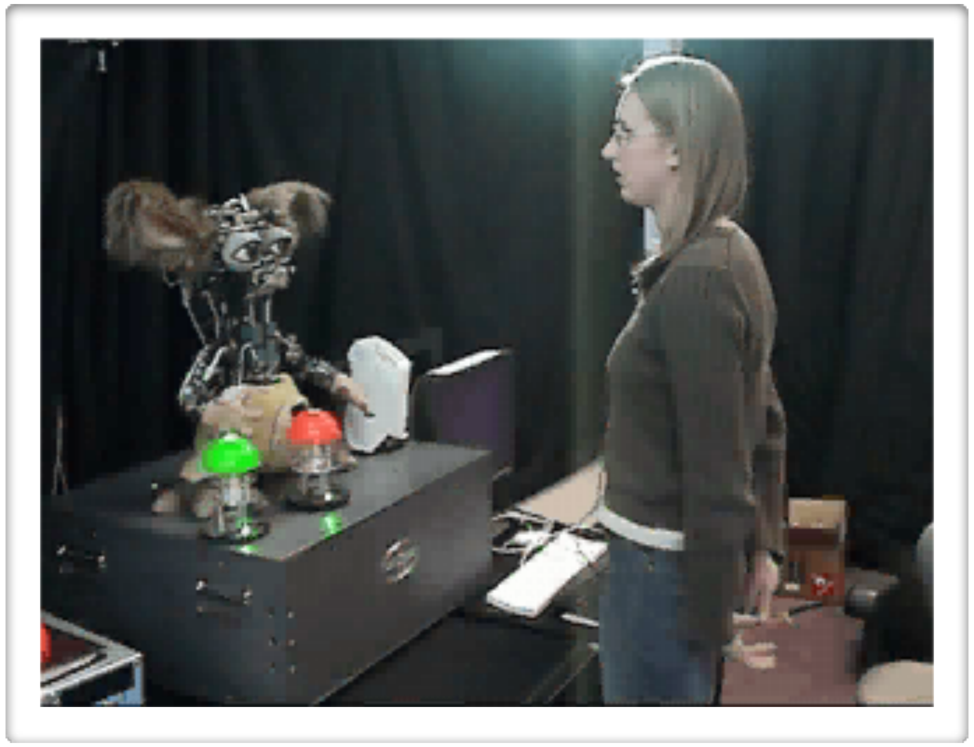
What is dialogue, anyway?

- Spoken dialogue is often *referential* to some (proximate or distant) spatio-temporal context
  - The context might be a small- or large-scale environment, a task to perform, a software application, etc.
  - *Grounding* dialogue thus requires the ability to resolve linguistic references to the (own) situation awareness and yield a common ground



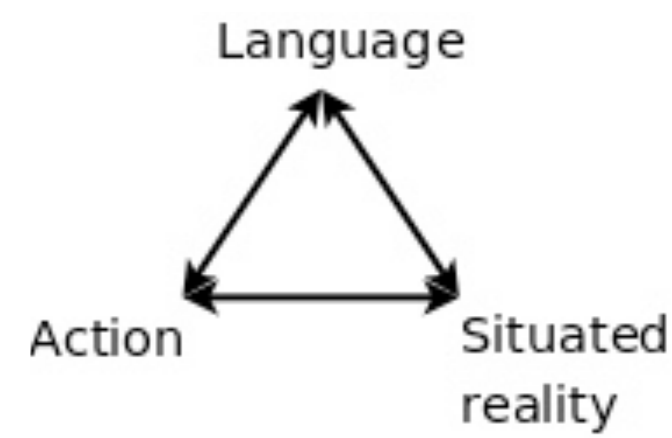
What is dialogue, anyway?

- Spoken dialogue is often *referential* to some (proximate or distant) spatio-temporal context
  - The context might be a small- or large-scale environment, a task to perform, a software application, etc.
  - *Grounding* dialogue thus requires the ability to resolve linguistic references to the (own) situation awareness and yield a common ground



Excursus: meaning in situated dialogue

- To acquire a real-world meaning, linguistic symbols must ultimately be grounded in other modalities
  - Connection to *perception* and *bodily experience*
  - For instance, the linguistic symbol "door" must somehow be connected to some prototypical *image* of a door
  - as well as to prototypical *affordances* (what can be typically done with a door, and how)
- For a more in-depth treatment of these questions in cognitive science, philosophy and artificial intelligence, a good starting reference:
  - "*Embodied Cognition: a Field Guide*" by Michael L. Anderson, Artificial Intelligence, 2003.



Standard problems to tackle

- The "usual" in spoken dialogue:
  - speech recognition errors
  - Partial, fragmentary, ungrammatical utterances,
  - Presence of many disfluencies (filled pauses, repairs, corrections, etc.)
  - Limited grammar coverage
  - Ambiguities at all processing levels
  - Uncertainty in contextual interpretation;
- Performance requirements for real-time dialogue
  - The system must be capable of responding *quickly* to any utterance, even in the presence of noisy, ambiguous, or distorted input

Disfluencies in spoken dialogue

Extract from a corpus of task-oriented, human-human spoken dialogue: *The Apollo Lunar Surface Journal.*

Parker : That's all we need. Go ahead and park on your 045 **<okay>**. We'll give you an update when you're done.
Cernan : Jack is **[it]** worth coming right there ?
Schmitt : **err** looks like a pretty **go/** good location.
Cernan : okay.
Schmitt : We can sample the rim materials of this crater. **(Pause)** Bob, I'm at the **uh** south **uh** let's say east-southeast rim of a, **oh**, 30-meter crater - **err** in the light mantle, of course - up on the **uh** Scarp and maybe 300...**(correcting himself) err** 200 meters from the **uh** rim of Lara in **(inaudible)** northeast direction.

Disfluencies in spoken dialogue

Extract from a corpus of task-oriented, human-human spoken dialogue: *The Apollo Lunar Surface Journal.*

Parker : That's all we need. Go ahead and park on your 045 **<okay>**. We'll give you an update when you're done.
Cernan : Jack is **[it]** worth coming right there ?
Schmitt : **err** looks like a pretty **go/** good location.
Cernan : okay.
Schmitt : We can sample the rim materials of this crater. **(Pause)** Bob, I'm at the **uh** south **uh** let's say east-southeast rim of a, **oh**, 30-meter crater - **err** in the light mantle, of course - up on the **uh** Scarp and maybe 300...**(correcting himself) err** 200 meters from the **uh** rim of Lara in **(inaudible)** northeast direction.

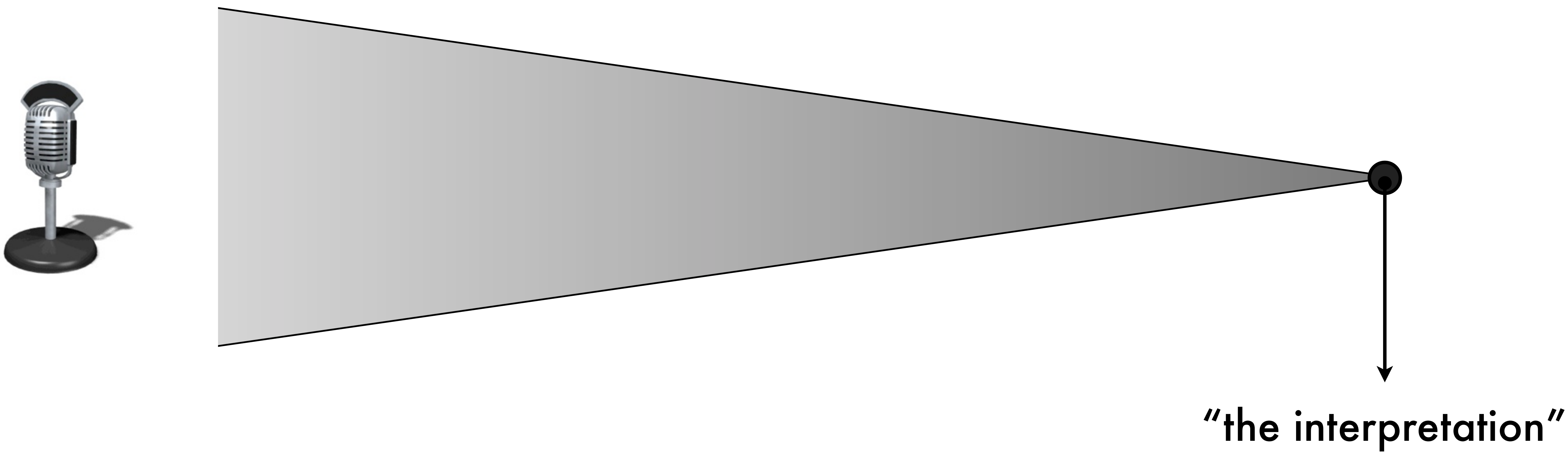
Robust processing of spoken dialogue

- How to improve the performance of dialogue understanding in such conditions?
- In my MSc thesis, I developed a new approach, based on the following strategy:
  - Improve the performance of speech recognition by exploiting *contextual knowledge* about the environment and the dialogue state.
  - Allow for a *controlled relaxation of the grammatical constraints* to account for spoken dialogue phenomena and speech recognition errors.
  - Finally, apply a *discriminative model* on the resulting set of interpretations, in order to select the most likely one given the context.
- We obtained very significant improvements in robustness and in accuracy compared to the baseline

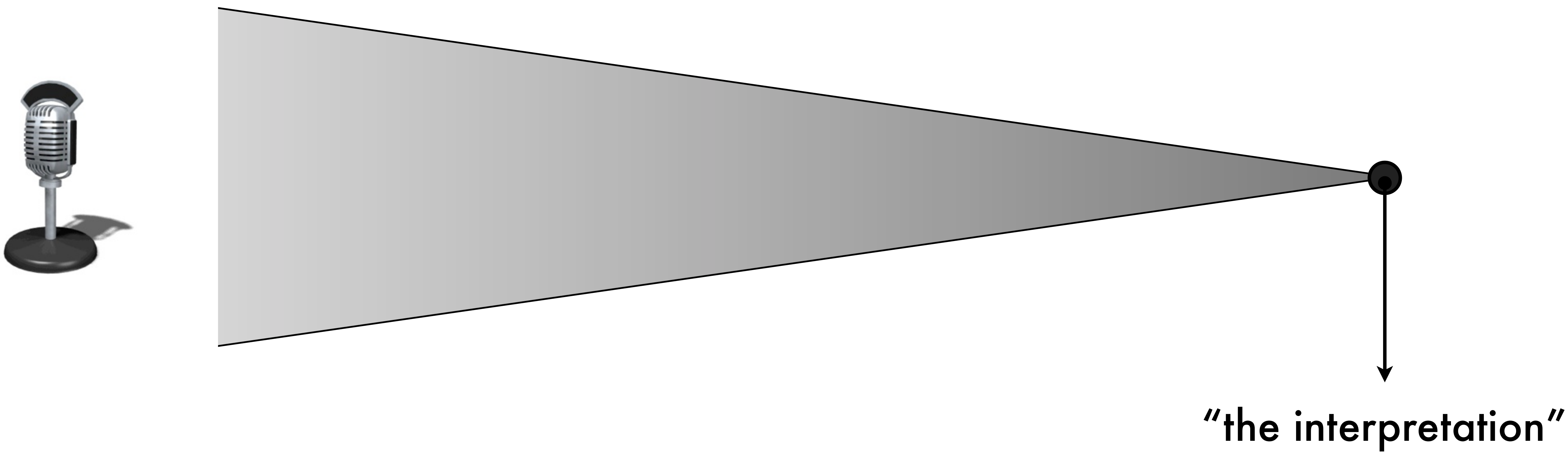
[Lison, P. 2008, 2009]

[Lison, P. & Kruijff G.-J. M 2009]

Robust processing of spoken dialogue

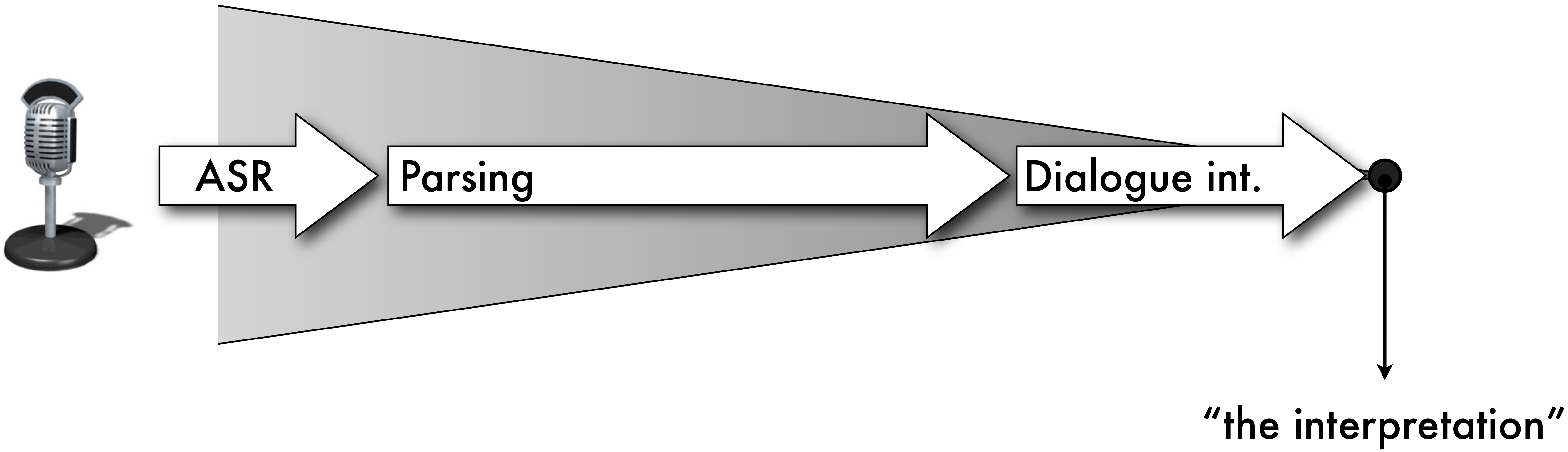


Robust processing of spoken dialogue



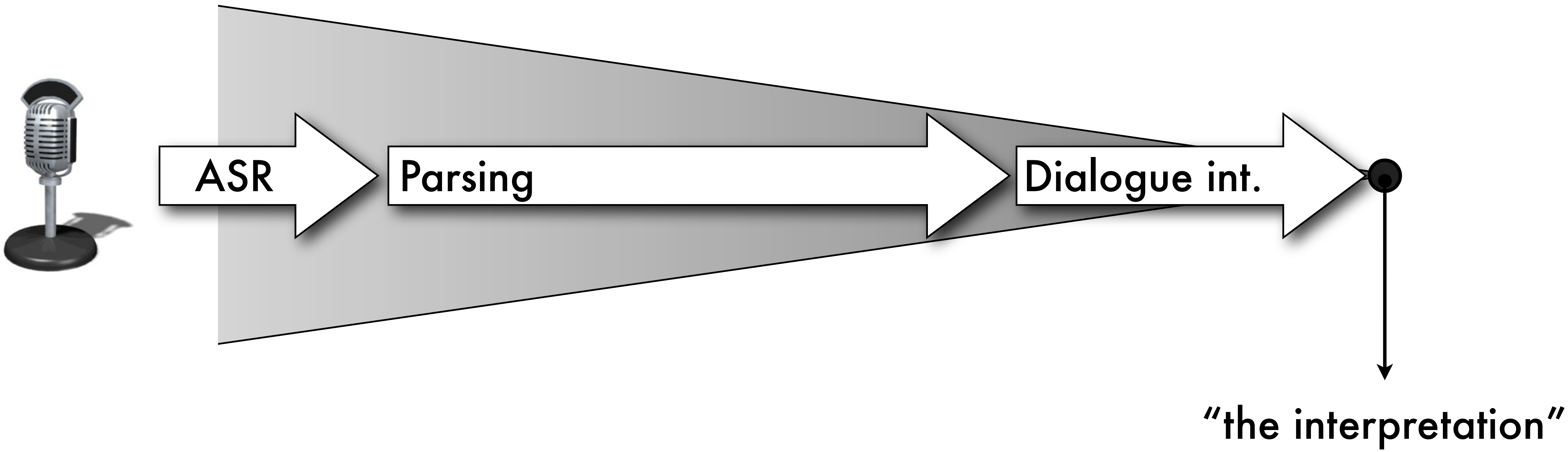
- Speech recognition with statistical models
- Incremental parsing with *Combinatory Categorial Grammar*
- Dialogue interpretation tasks: reference resolution, dialogue move recognition, etc.

Robust processing of spoken dialogue



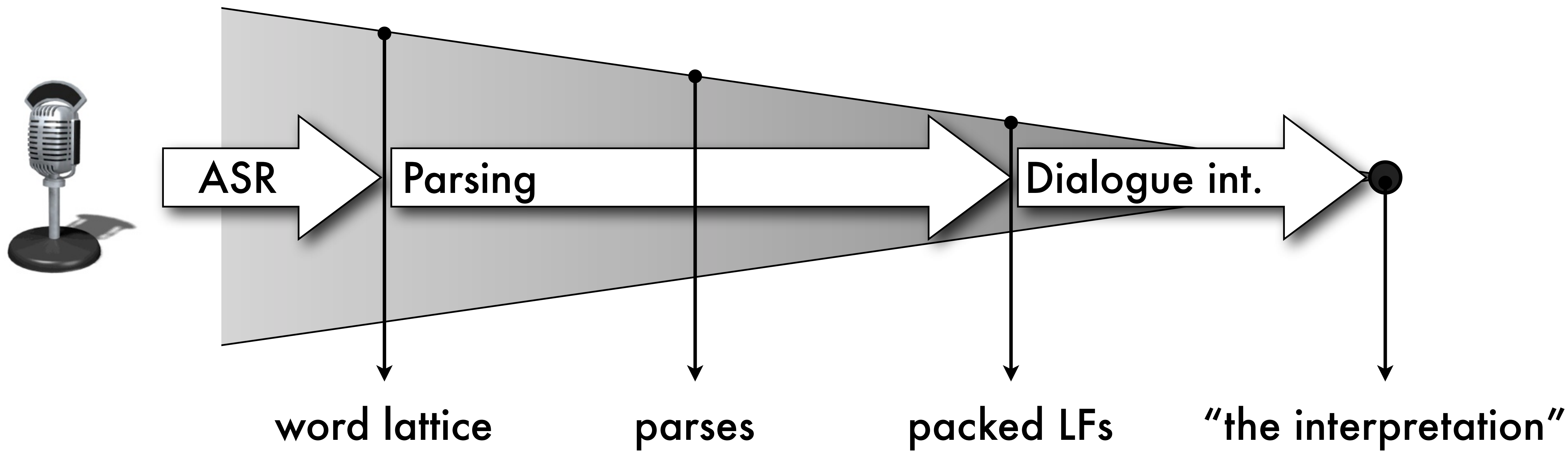
- Speech recognition with statistical models
- Incremental parsing with *Combinatory Categorial Grammar*
- Dialogue interpretation tasks: reference resolution, dialogue move recognition, etc.

Robust processing of spoken dialogue



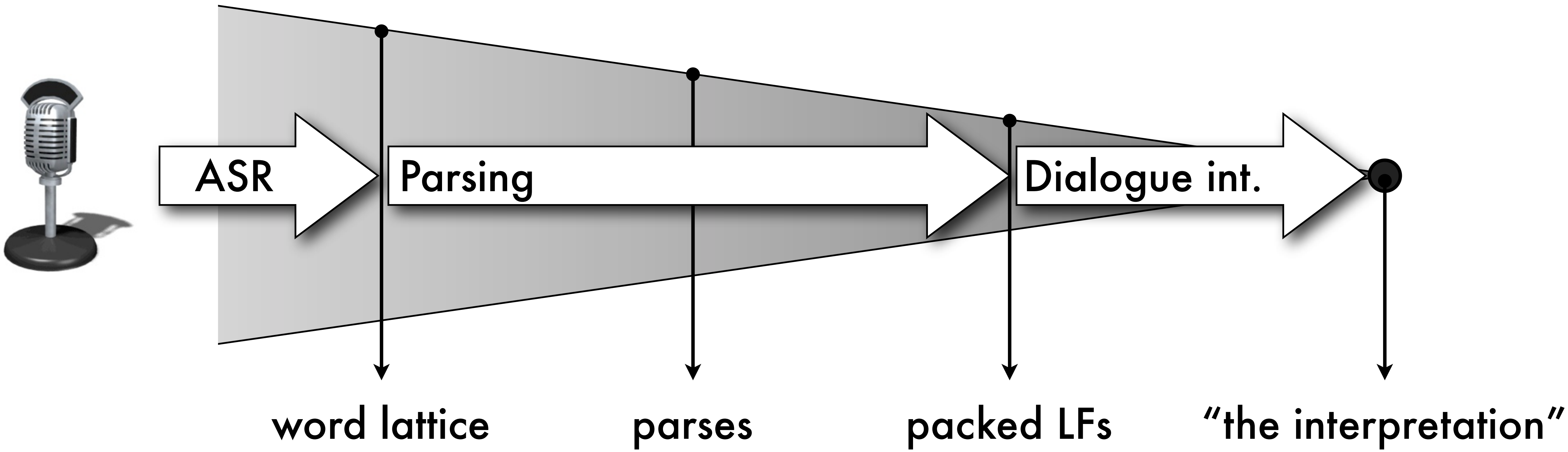
- Speech recognition outputs a *word lattice*
  - Word lattice = set of alternative recognition hypotheses compacted in a directed graph
- The CCG parser takes a *word lattice* as input and outputs partial *semantic representations* (logical forms)
  - Logical forms are expressed as ontologically richly sorted, relational structures
- Dialogue interpretation based on dialogue structure

Robust processing of spoken dialogue



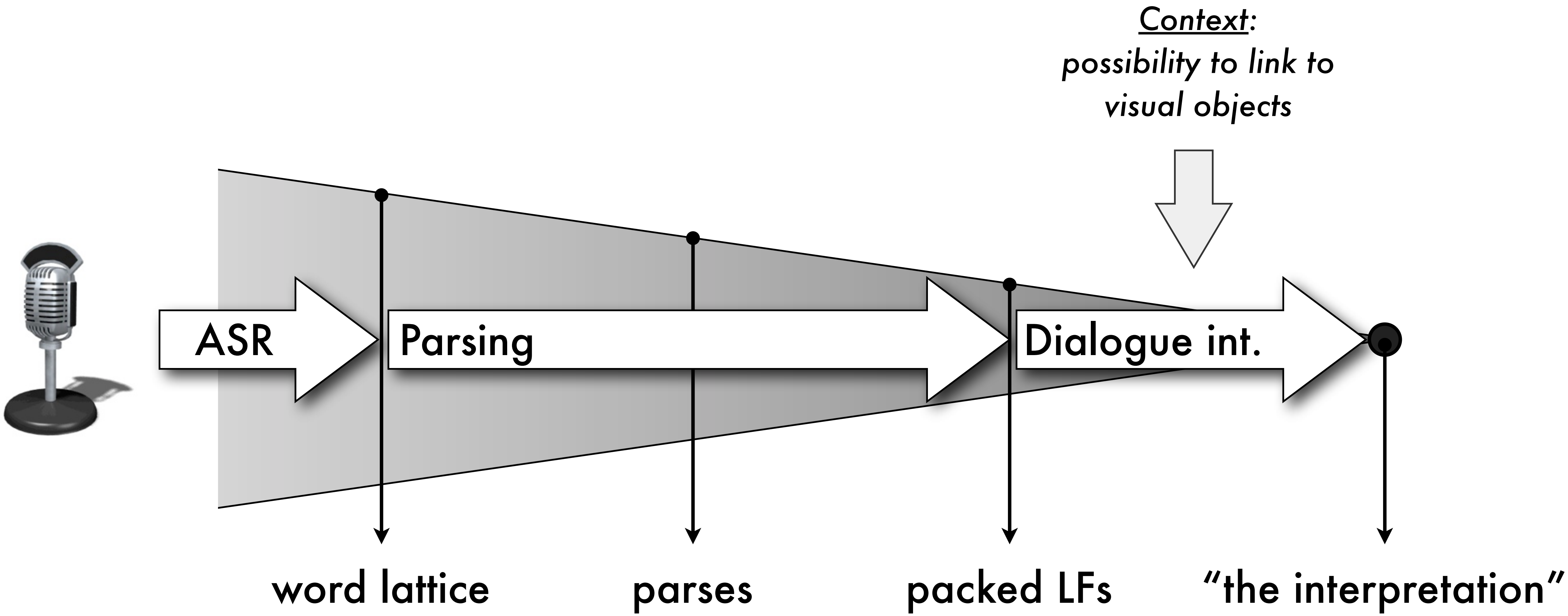
- Speech recognition outputs a *word lattice*
  - Word lattice = set of alternative recognition hypotheses compacted in a directed graph
- The CCG parser takes a *word lattice* as input and outputs partial *semantic representations* (logical forms)
  - Logical forms are expressed as ontologically richly sorted, relational structures
- Dialogue interpretation based on dialogue structure

Robust processing of spoken dialogue



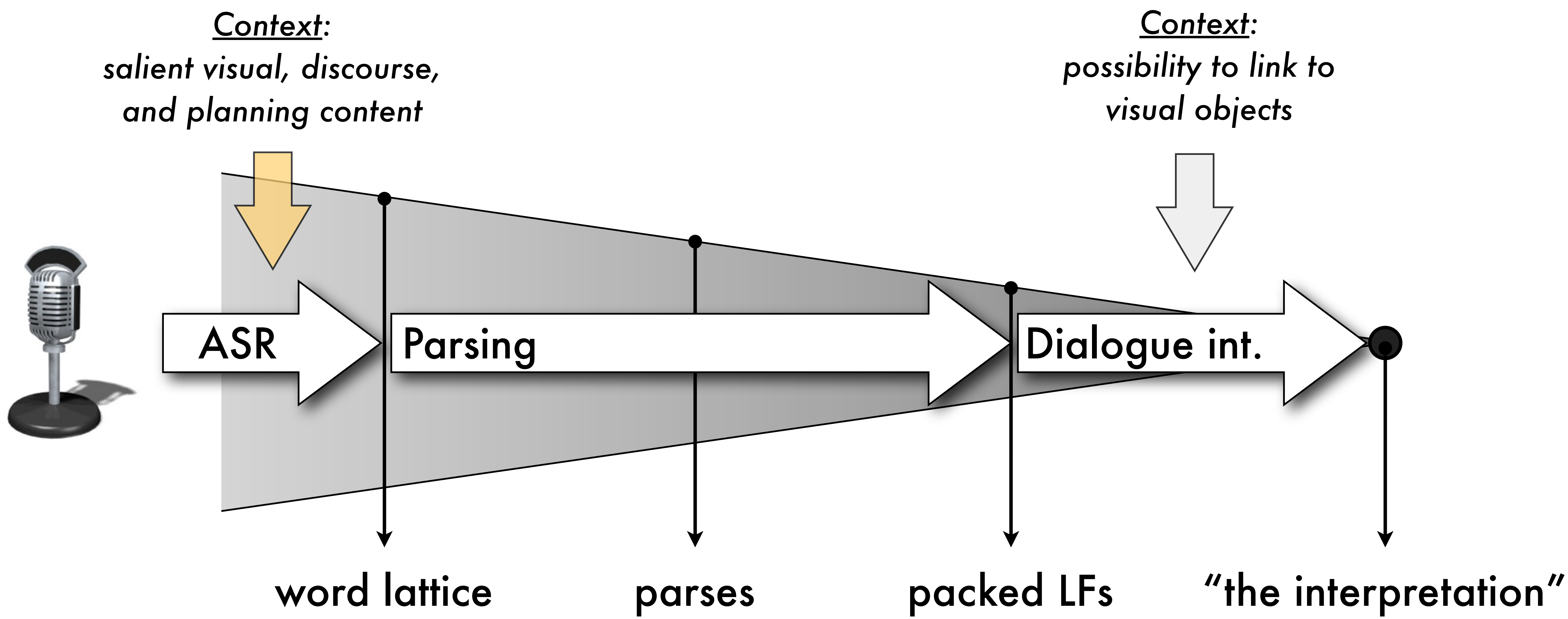
- *Context* exploitation to prime ASR language models
- *Controlled relaxation* the grammatical constraints to handle ill-formed or misrecognised utterances
- Finally, use of a *discriminative parse selection* model to select the best analysis among the possible ones

Robust processing of spoken dialogue



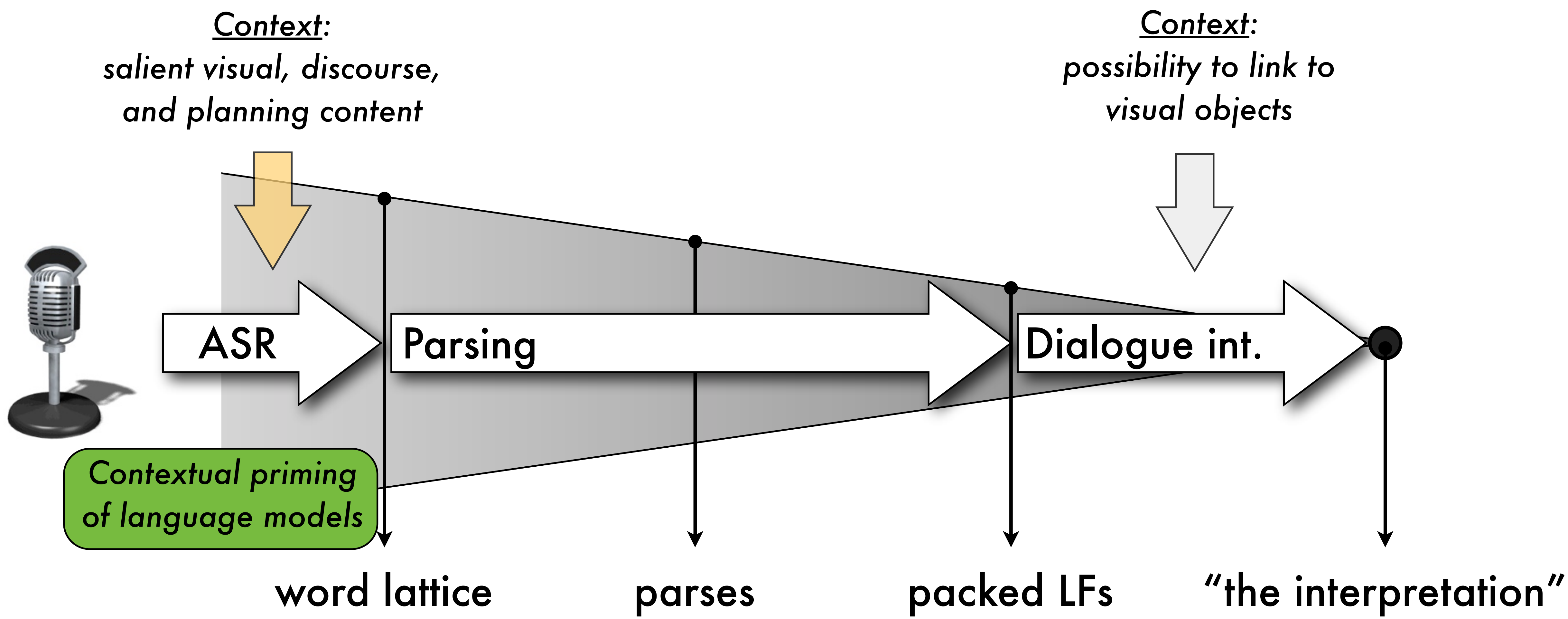
- *Context* exploitation to prime ASR language models
- *Controlled relaxation* the grammatical constraints to handle ill-formed or misrecognised utterances
- Finally, use of a *discriminative parse selection* model to select the best analysis among the possible ones

Robust processing of spoken dialogue



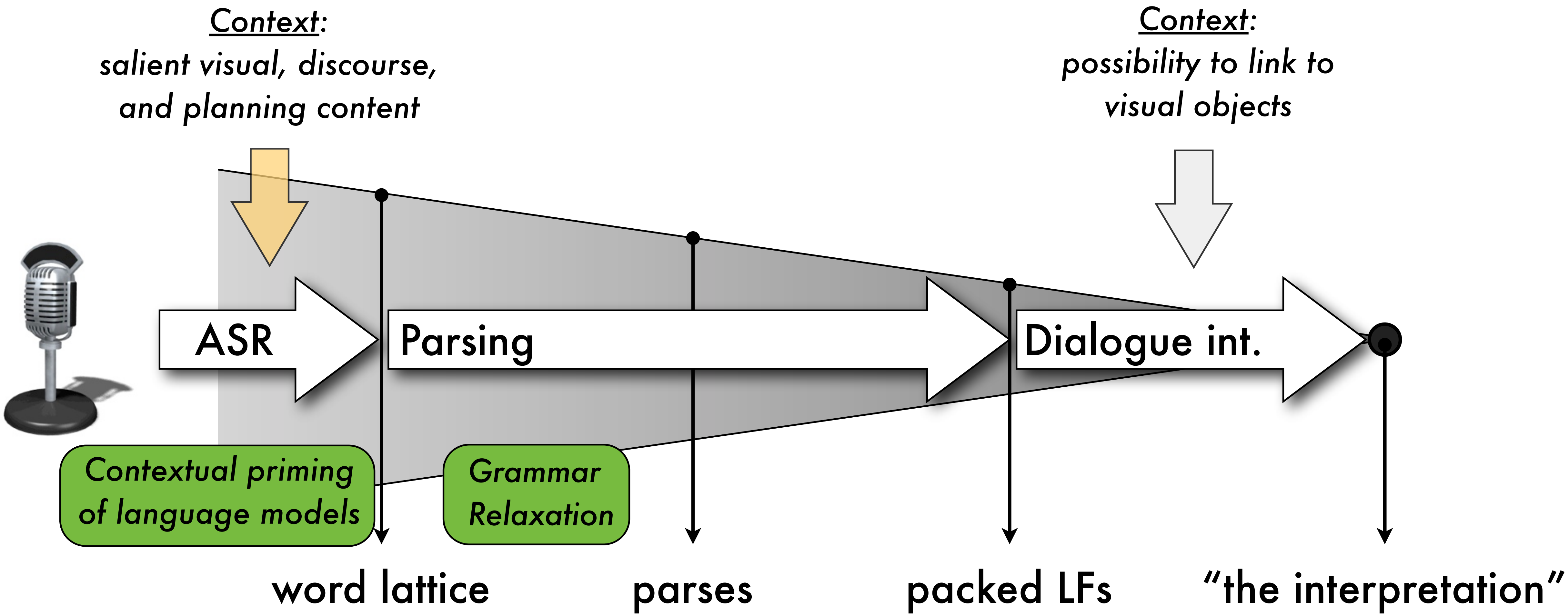
- *Context* exploitation to prime ASR language models
- *Controlled relaxation* the grammatical constraints to handle ill-formed or misrecognised utterances
- Finally, use of a *discriminative parse selection* model to select the best analysis among the possible ones

Robust processing of spoken dialogue



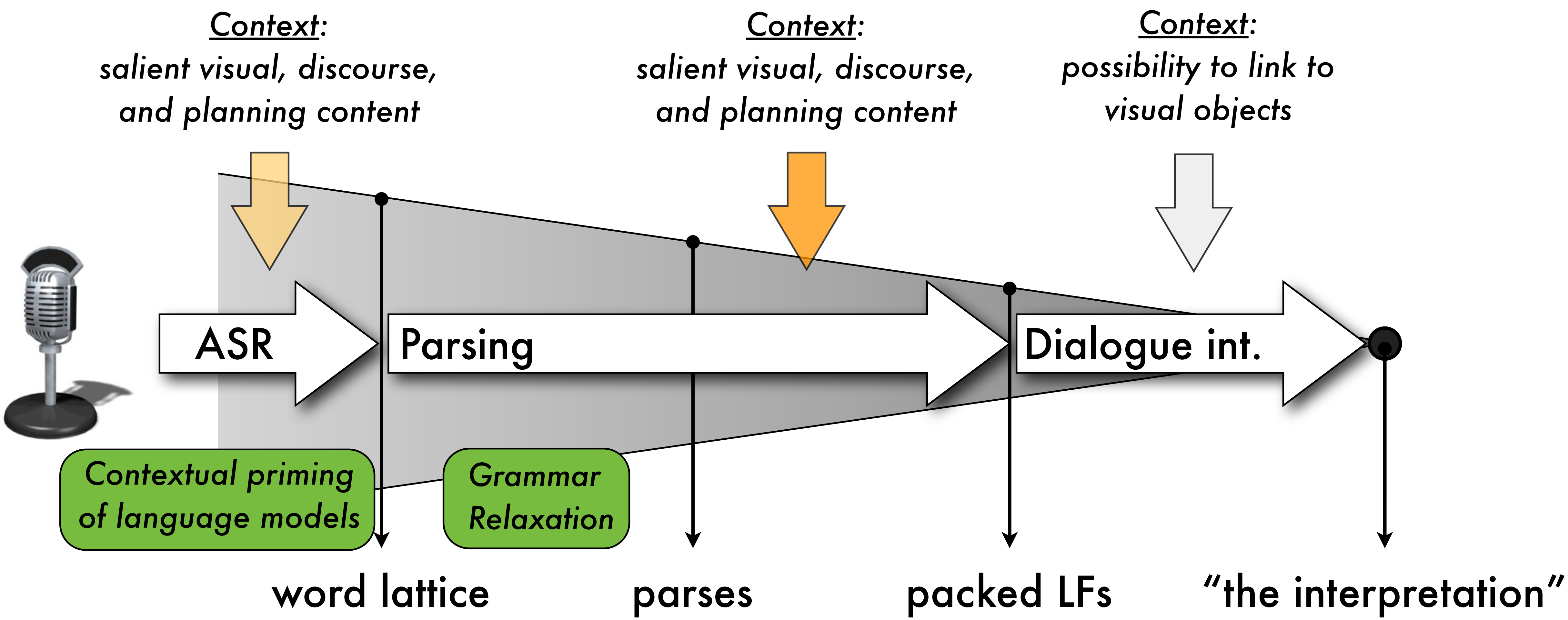
- *Context* exploitation to prime ASR language models
- *Controlled relaxation* the grammatical constraints to handle ill-formed or misrecognised utterances
- Finally, use of a *discriminative parse selection* model to select the best analysis among the possible ones

Robust processing of spoken dialogue



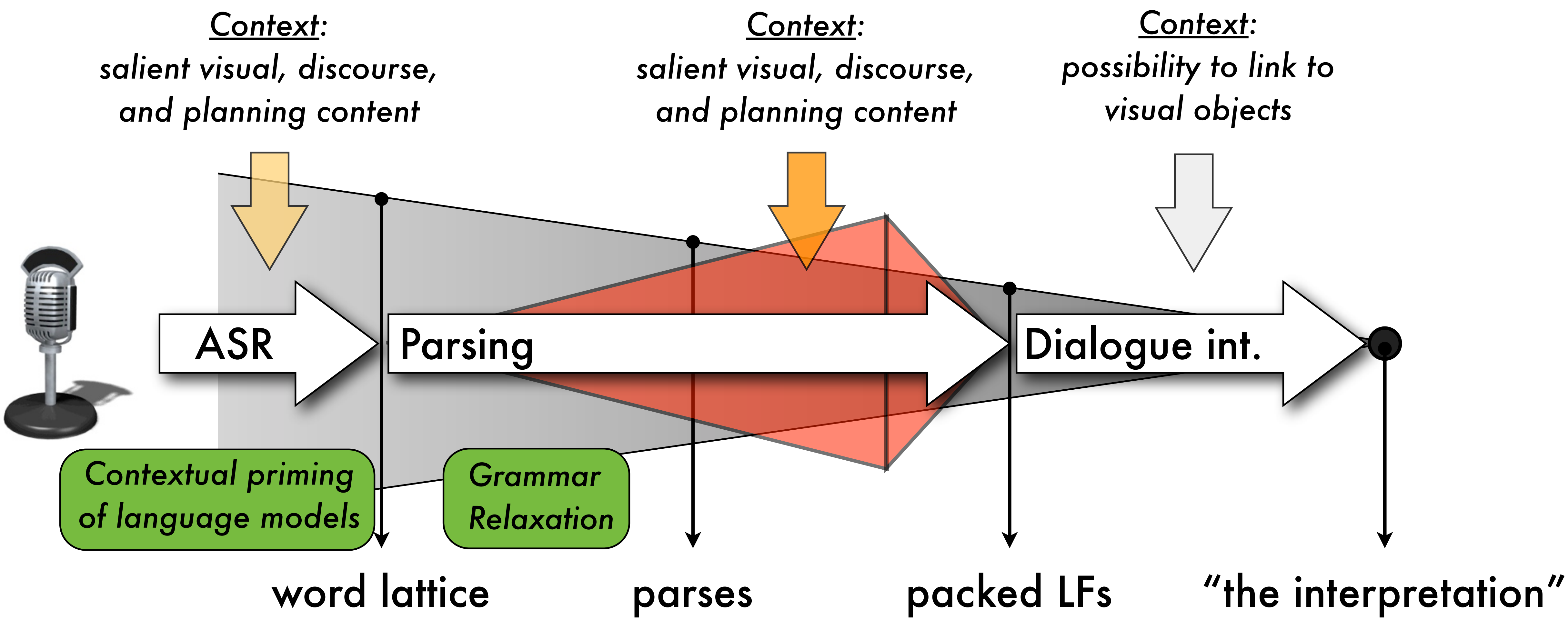
- *Context* exploitation to prime ASR language models
- *Controlled relaxation* the grammatical constraints to handle ill-formed or misrecognised utterances
- Finally, use of a *discriminative parse selection* model to select the best analysis among the possible ones

Robust processing of spoken dialogue



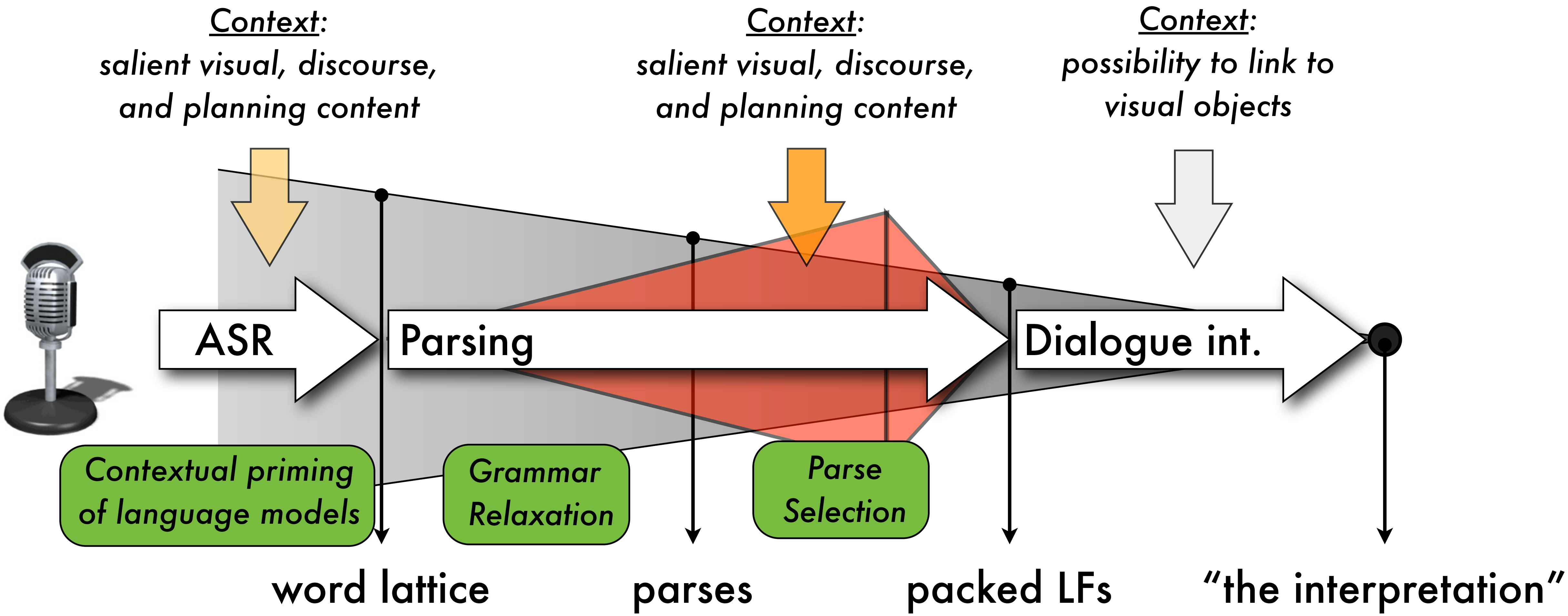
- *Context* exploitation to prime ASR language models
- *Controlled relaxation* the grammatical constraints to handle ill-formed or misrecognised utterances
- Finally, use of a *discriminative parse selection* model to select the best analysis among the possible ones

Robust processing of spoken dialogue



- *Context* exploitation to prime ASR language models
- *Controlled relaxation* the grammatical constraints to handle ill-formed or misrecognised utterances
- Finally, use of a *discriminative parse selection* model to select the best analysis among the possible ones

Robust processing of spoken dialogue

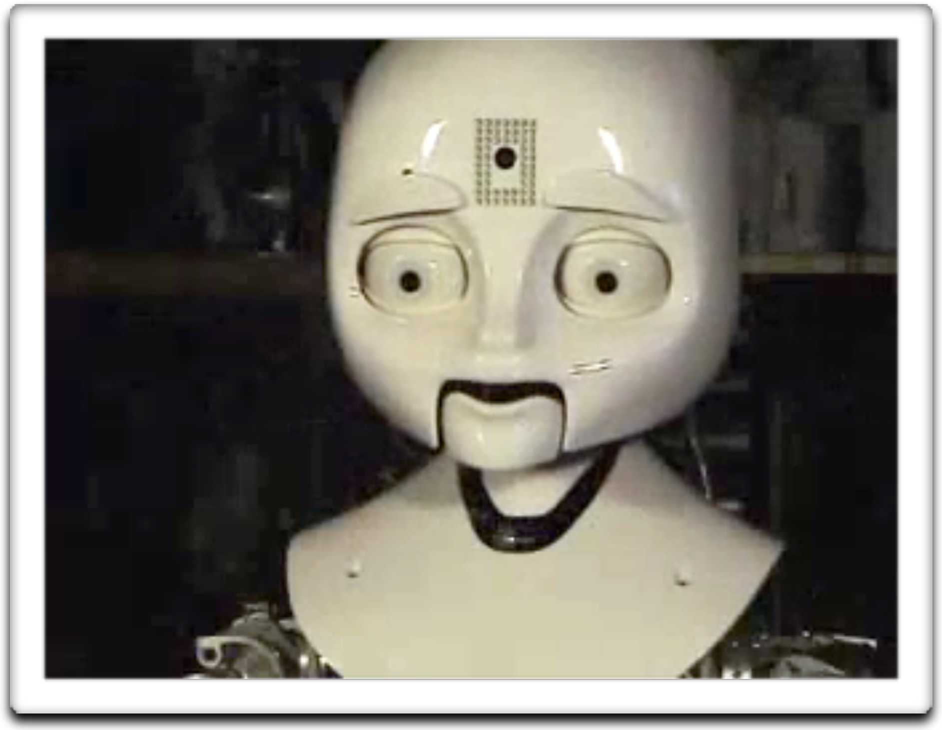


- *Context* exploitation to prime ASR language models
- *Controlled relaxation* the grammatical constraints to handle ill-formed or misrecognised utterances
- Finally, use of a *discriminative parse selection* model to select the best analysis among the possible ones

Excursus: non-verbal interaction

- **Non-verbal interaction** plays a crucial role in situated interactions
  - Gestures, posture, affective display
- Talking robots should perceive the body language of their interlocutor, and use their own body for interacting as well
  - Both non-verbal language *understanding* and *production*
  - Humans will naturally expect the robot to behave in such a way, and will perceive the interaction as awkward if it doesn't

Kismet, MIT Media Lab



Dexter, MIT Media Lab

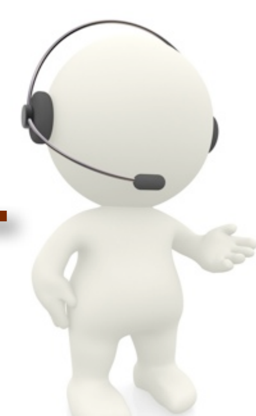
Dialogue systems architecture

Dialogue systems architecture

User

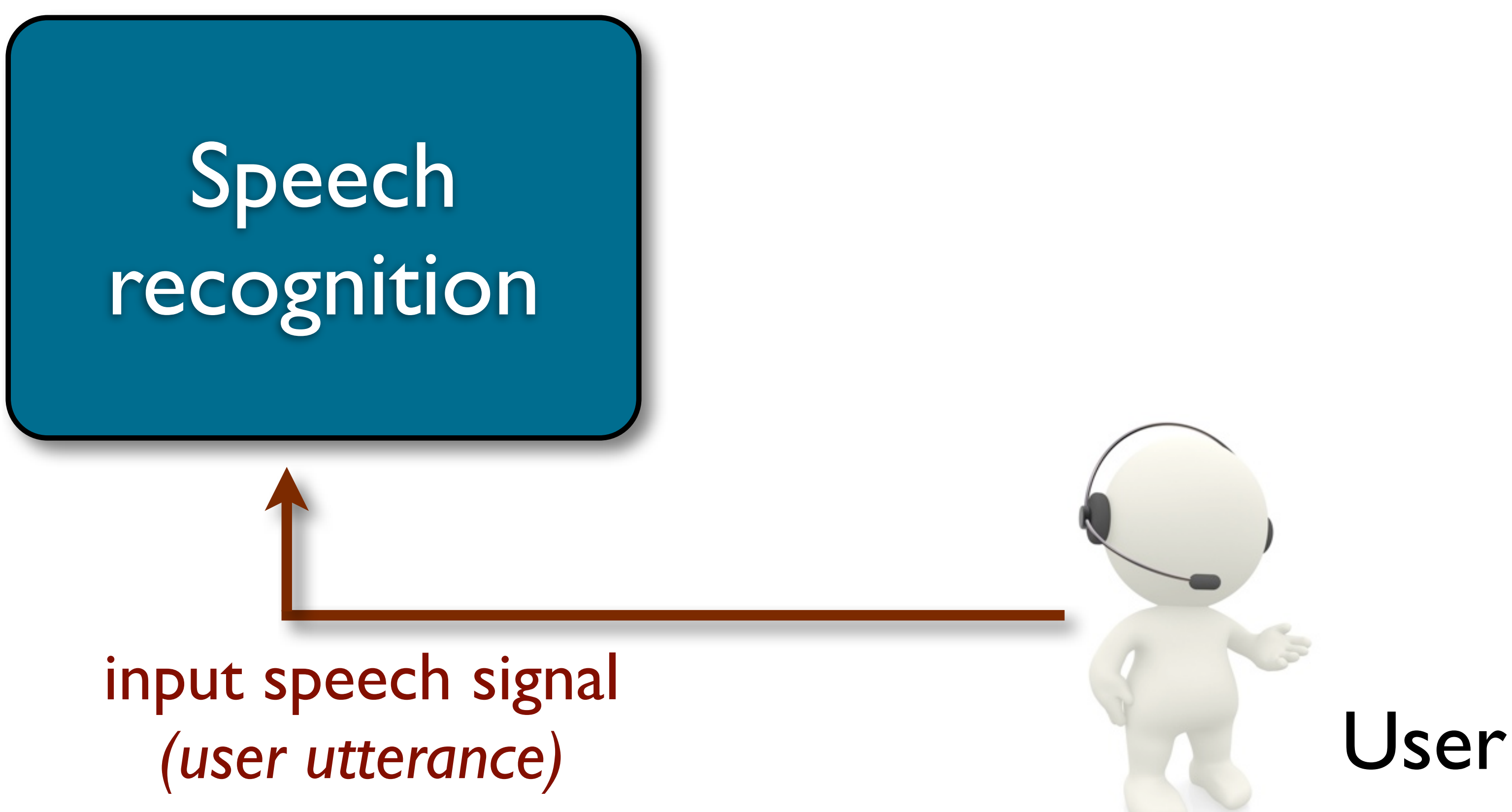
Dialogue systems architecture

input speech signal
(user utterance)

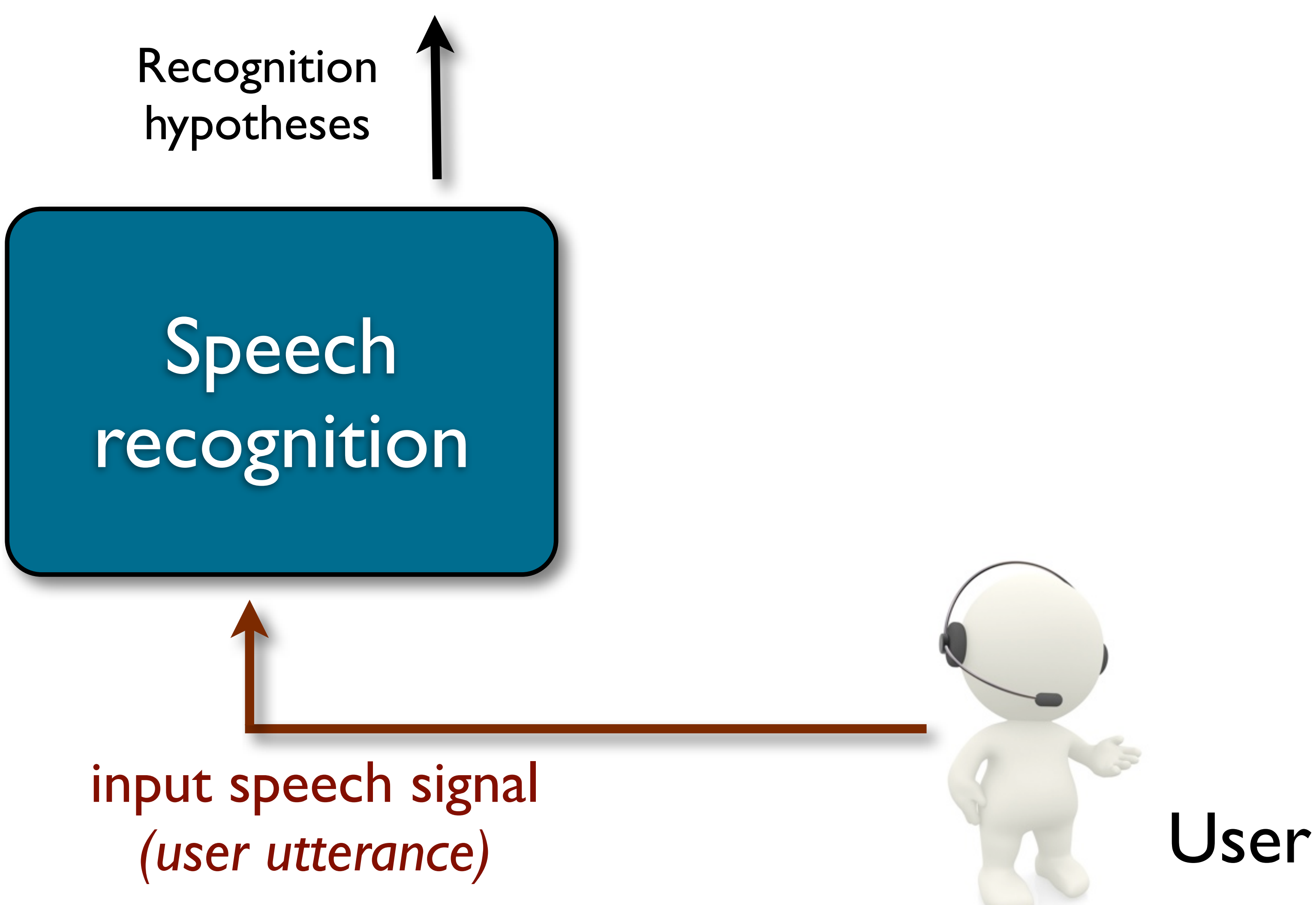


User

Dialogue systems architecture



Dialogue systems architecture



Dialogue systems architecture

Speech understanding

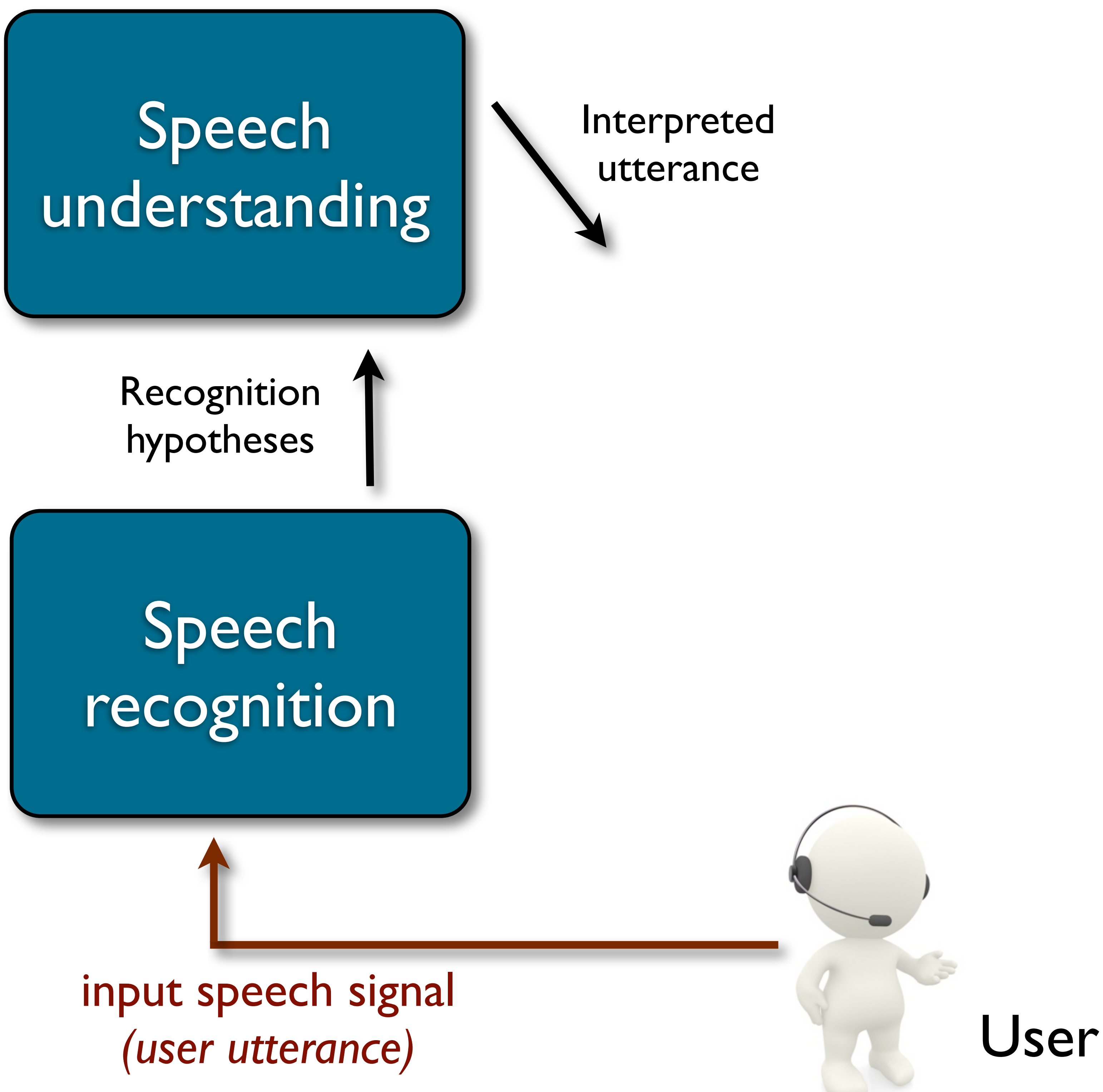
Recognition hypotheses

Speech recognition

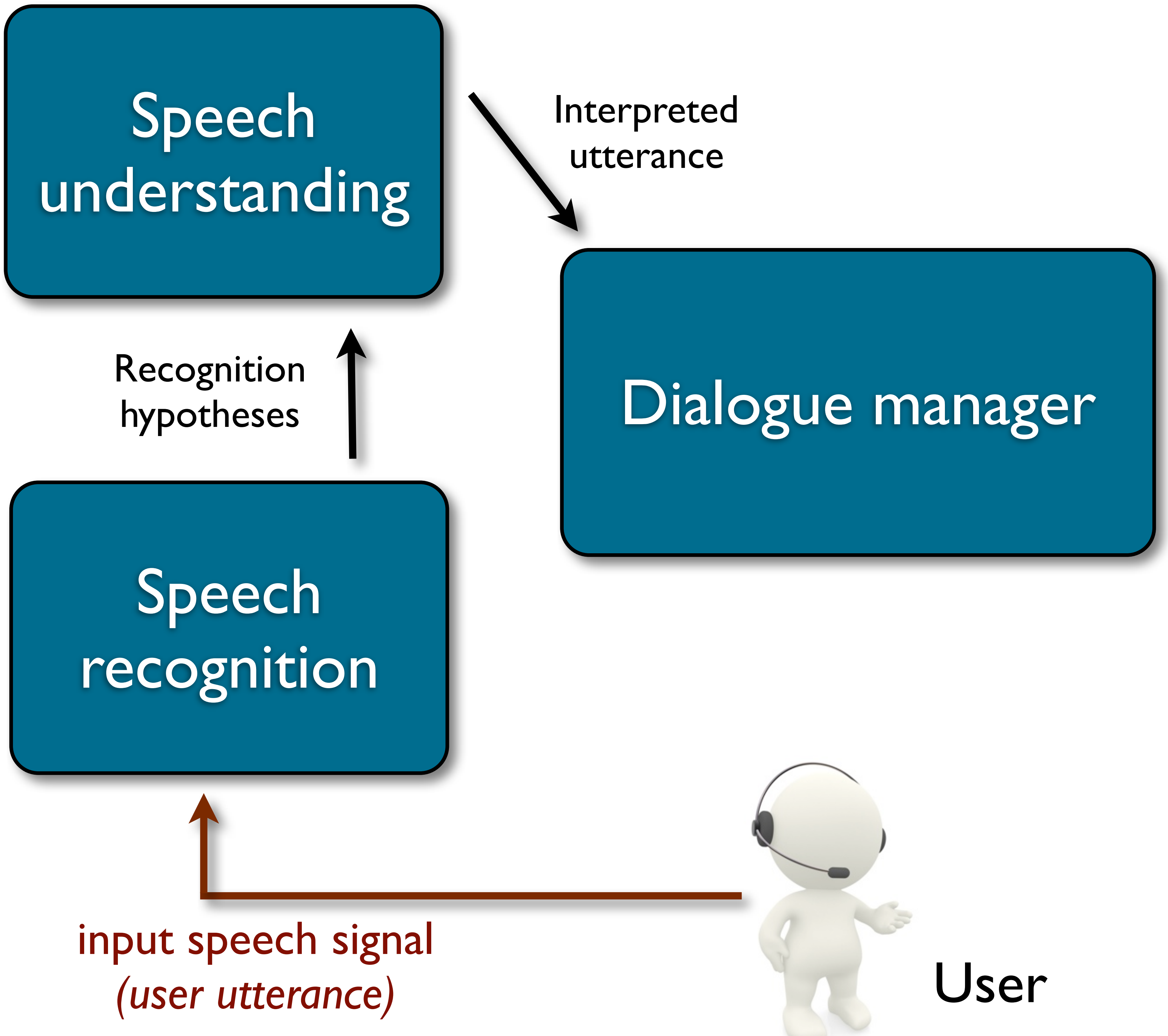
input speech signal
(user utterance)

User

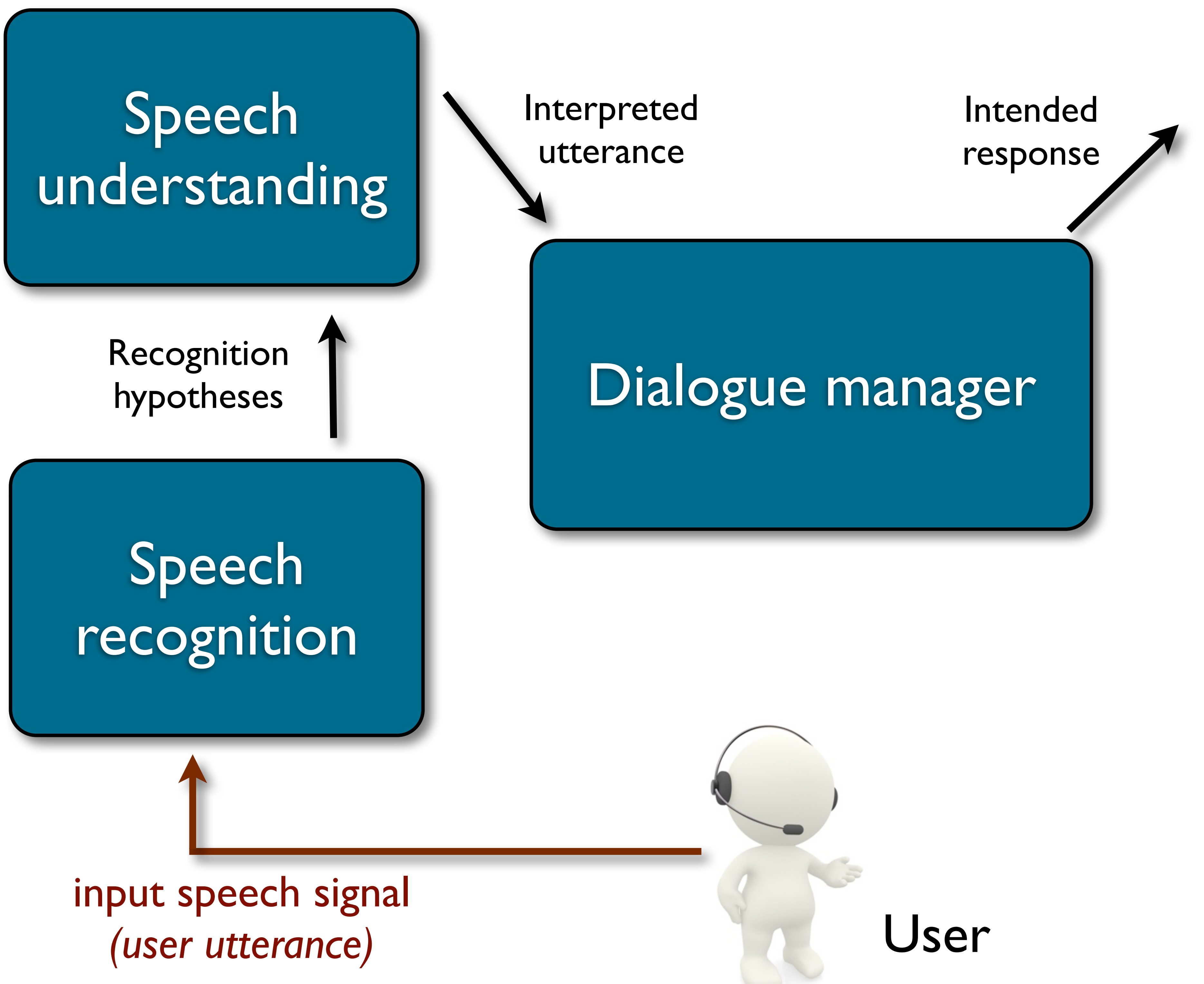
Dialogue systems architecture



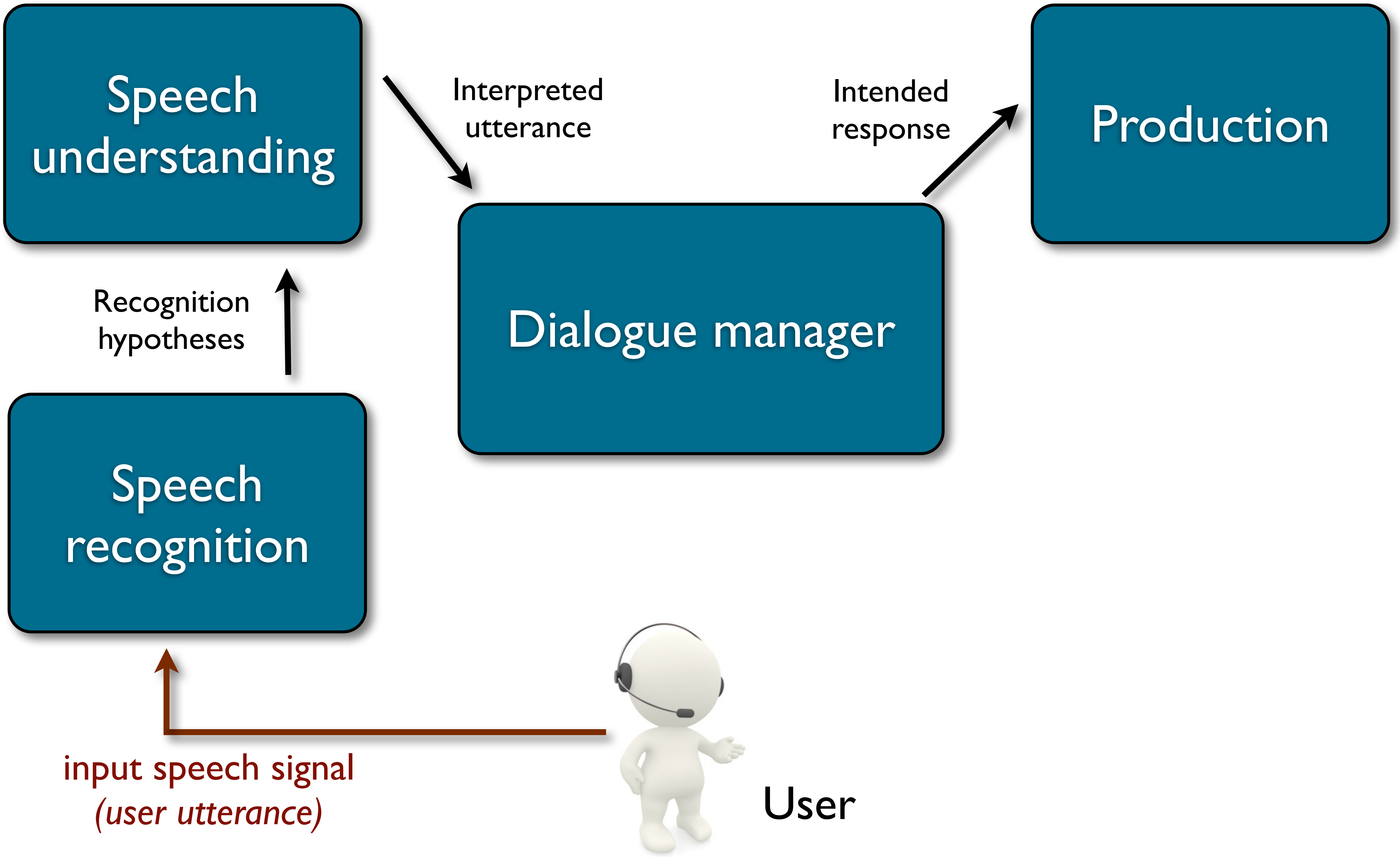
Dialogue systems architecture



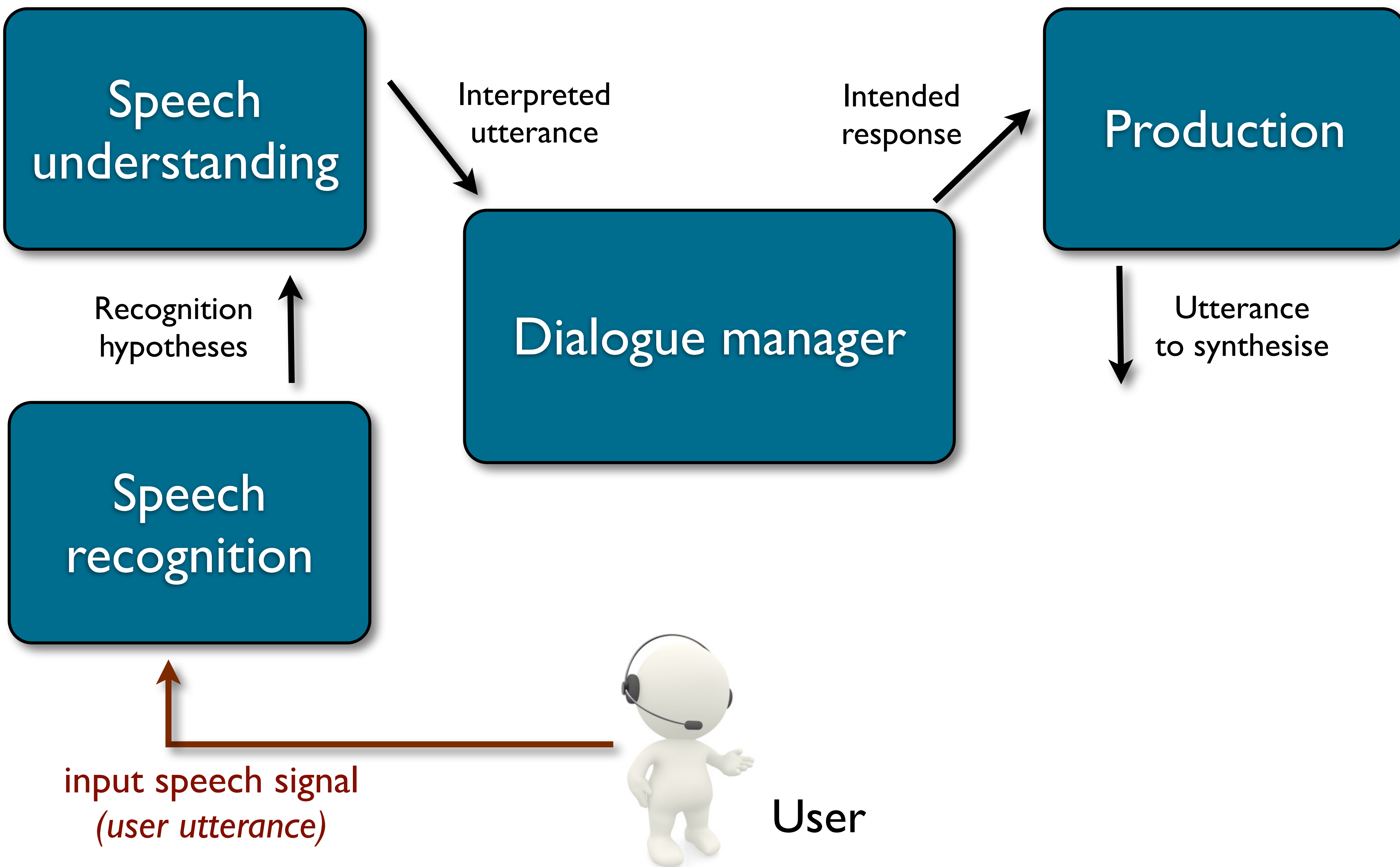
Dialogue systems architecture



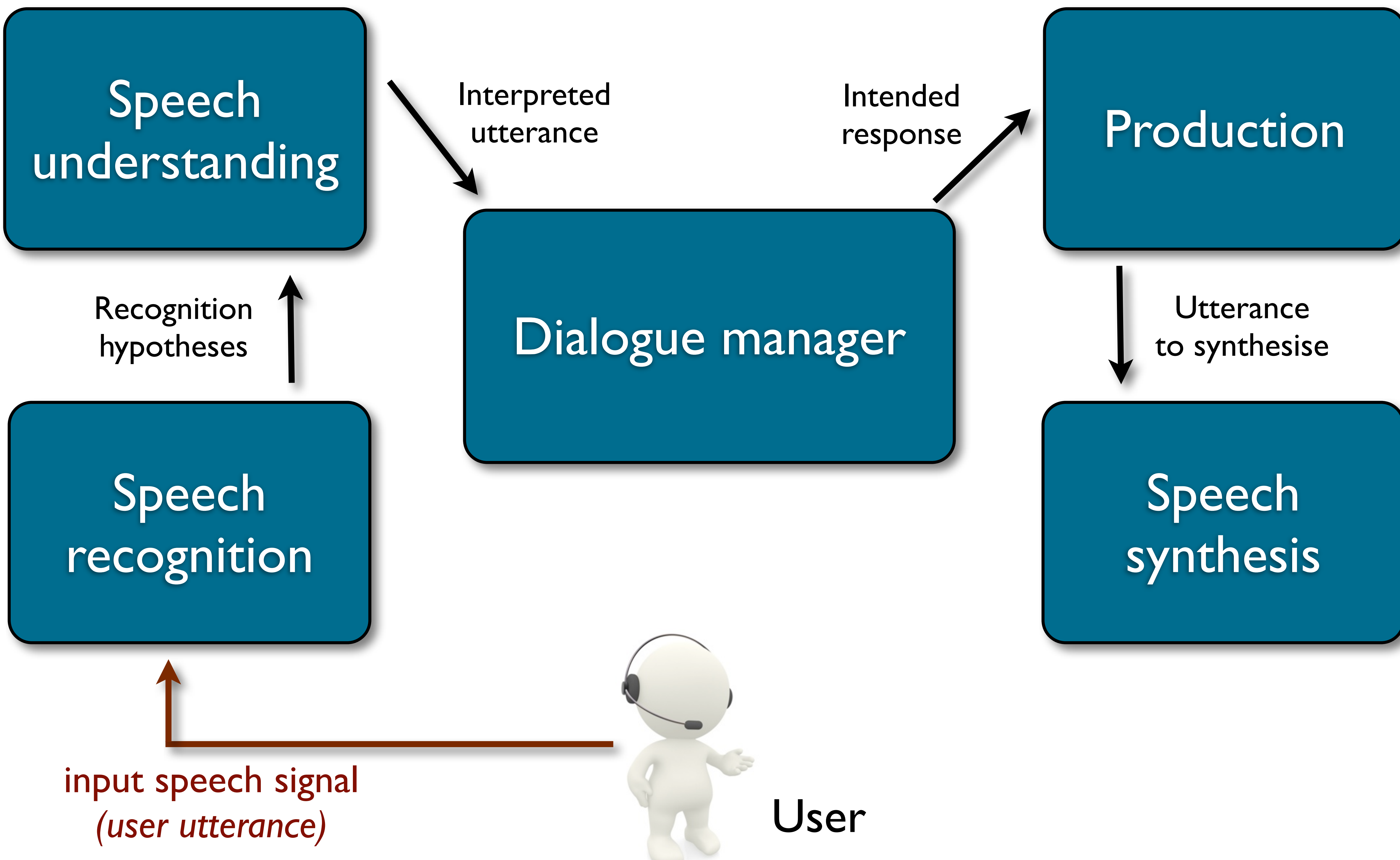
Dialogue systems architecture



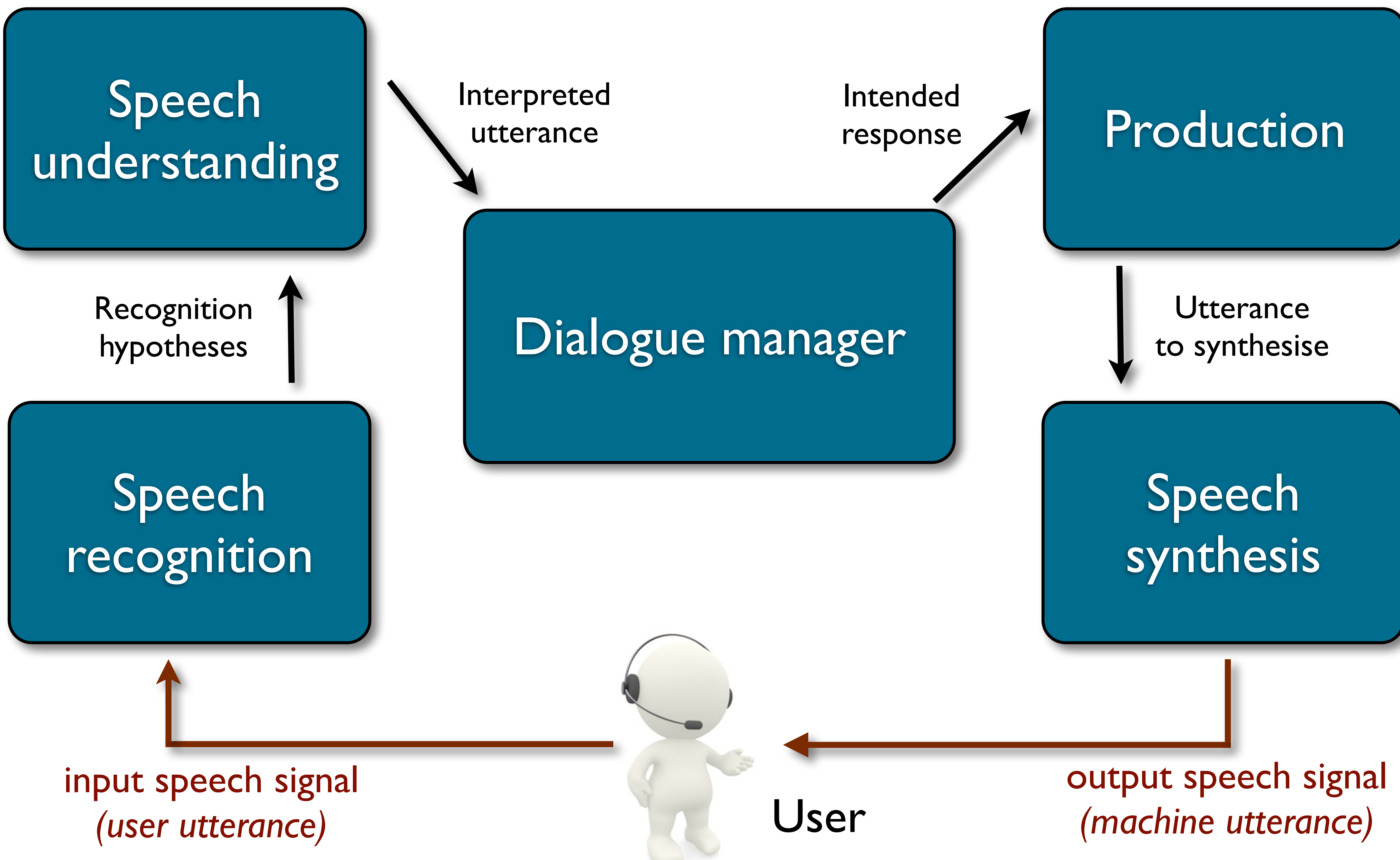
Dialogue systems architecture



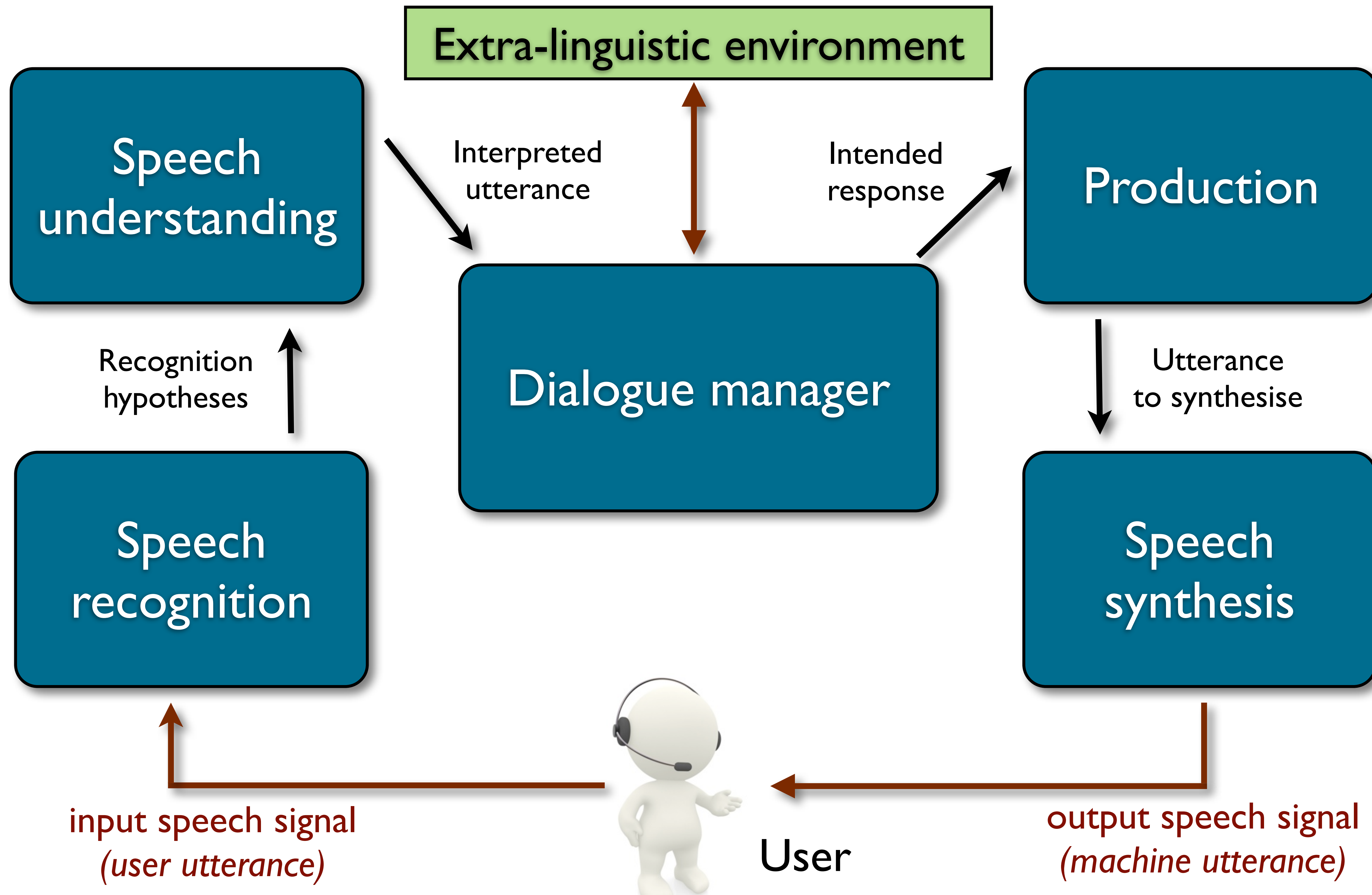
Dialogue systems architecture



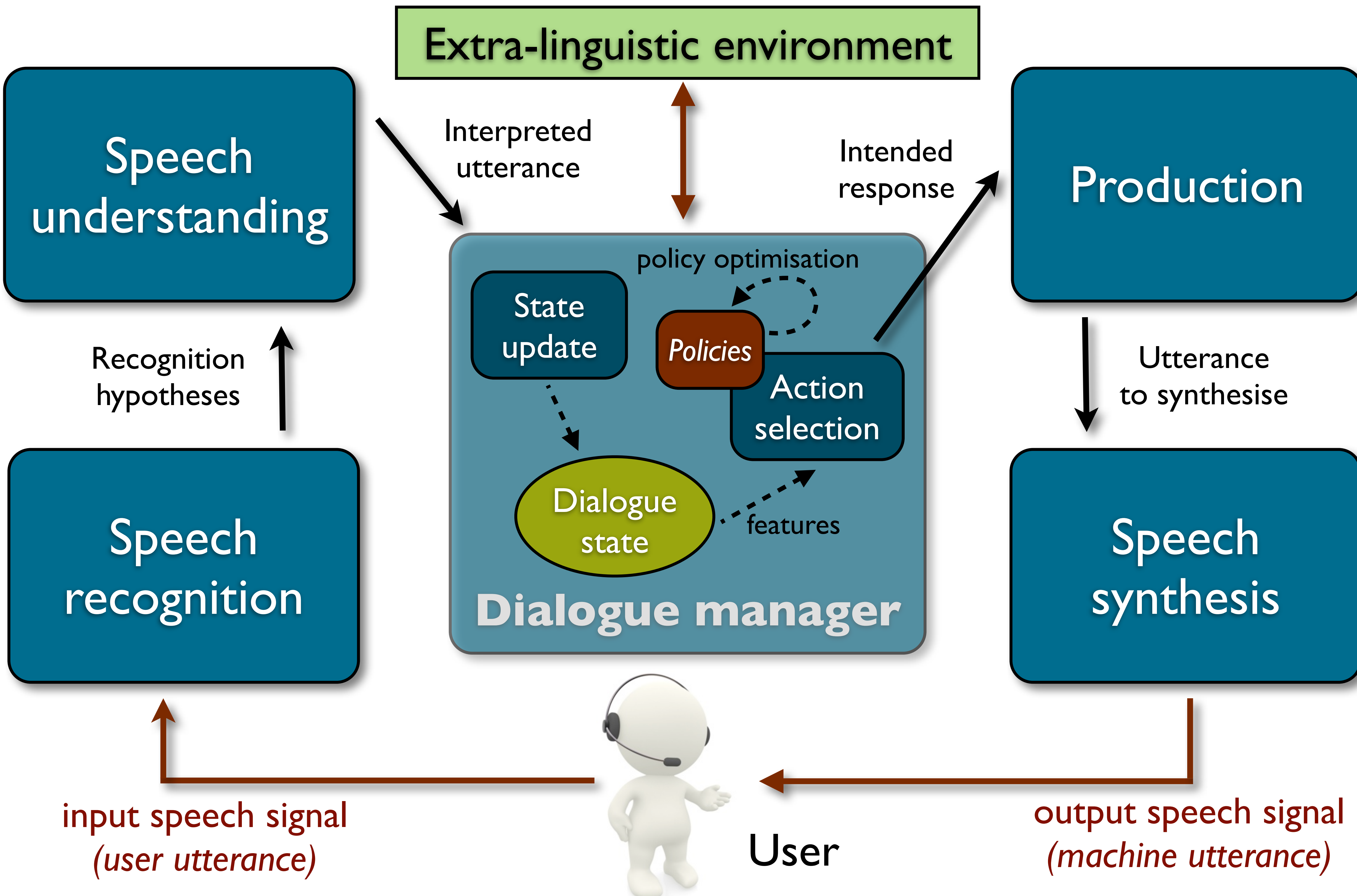
Dialogue systems architecture



Dialogue systems architecture



Dialogue systems architecture

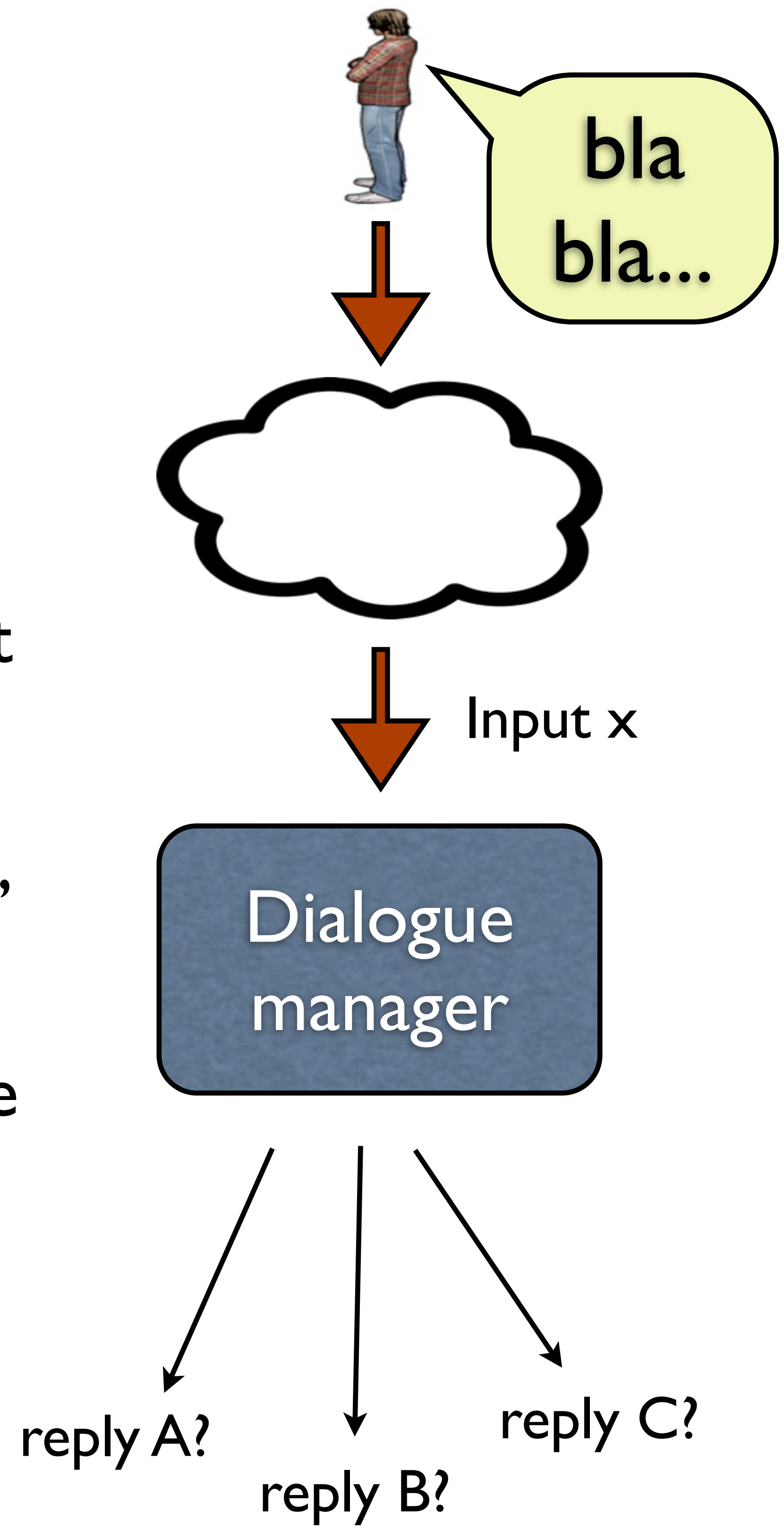


Outline of the talk

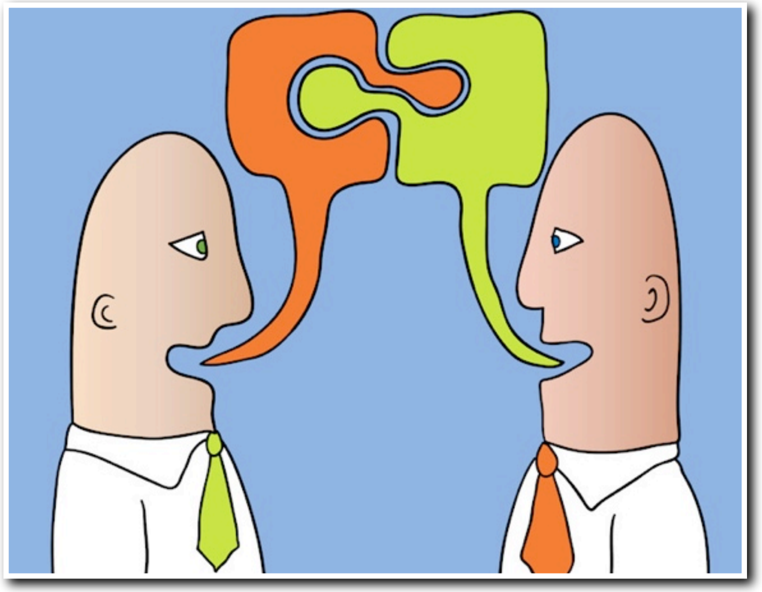
- Background
  - Generalities about dialogue systems
  - **Dialogue management, dialogue models**
  - Learning dialogue policies
- Ongoing work
  - General vision
  - Current research directions
  - Implementation & evaluation
- Conclusion

Dialogue management

- Main focus on the **dialogue management** (DM) part of spoken dialogue systems
- Dialogue management is all about **decision-making**
  - i.e. what should the system decide to *say* or *do* at a given point of an interaction
  - more precisely: decision-making *under uncertainty*, since the communication channel is noisy
  - The action set available to the system can include both linguistic and non-linguistic actions
  - The same holds for the observation set (e.g. multimodal interaction)



Functionalities of DM



- The dialogue manager is thus responsible for controlling the *flow* of the interaction
- Human conversational skills that the dialogue system should seek to emulate:
    - Interpret utterances *contextually*;
    - Recognise the hidden *structure* and *function* of the interaction;
    - Manage *turn-taking*;
    - Fulfill conversational *obligations* & social *conventions*;
    - *Plan* multi-utterance responses;
    - Manage the system *uncertainty*

Dialogue policies

- Dialogue management decisions are usually pre-encoded in so-called dialogue **policies**
  - Dialogue policies specify how the system should react for every possible state
  - They can be either *manually encoded* by the system designer, or *learned* via machine learning techniques (more on this in a few slides)
- Policies are typically designed/learned *off-line*, in order to minimize the computations necessary at runtime
  - In which case the only runtime task of the dialogue manager is to determine which state it is exactly in
  - There is also some work on the machine learning literature on hybrid algorithms combining both *offline* and *online* planning

[S. Ross, J. Pineau, S. Paquet, B. Chaib-draa 2008]

The dialogue management problem

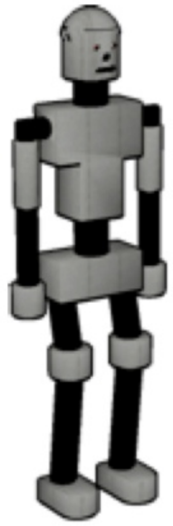
- Dialogue management as a planning/control problem:
  - The dialogue system is an artificial agent
  - which is able to perceive its environment *(e.g. user utterances)*
  - and which is able to perform actions *(e.g. system responses)*
  - to perform one or several goals *(the application purpose)*
  - given some task-specific costs and constraints
- There are many possible paths leading towards the goal
- And complex trade-offs to determine

The dialogue management problem

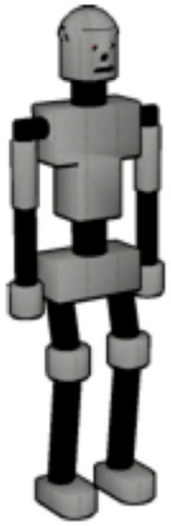
- But dialogue management is slightly different from classical planning domains:
  - There is usually no *complete* model of the interaction (specifying all possible dialogue paths)
  - The present interaction state is often only *partially observable*
  - The systems goals are *multiple* and *mutually competing*
  - There is usually *very little time* available at runtime for perform full planning (soft real-time reactivity)
- That's why *reinforcement learning* techniques are often preferred to classical planning in dialogue management

Dialogue management in HRI

Dialogue management in HRI



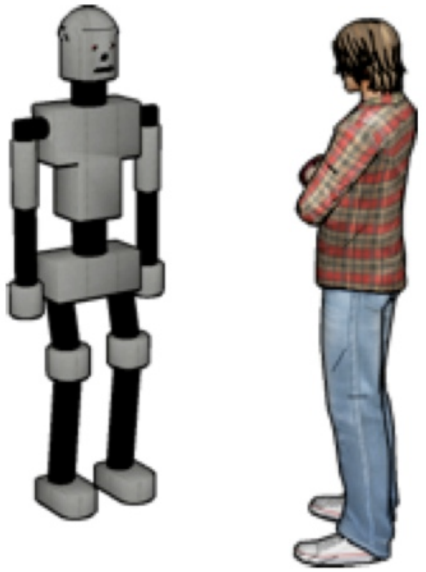
Dialogue management in HRI



A robot...

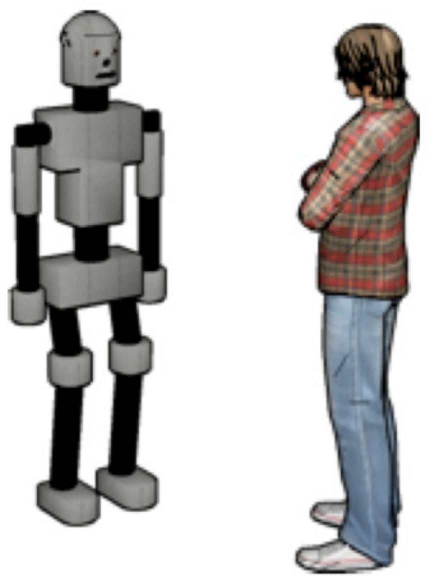
UiO : University of Oslo

Dialogue management in HRI



A robot...

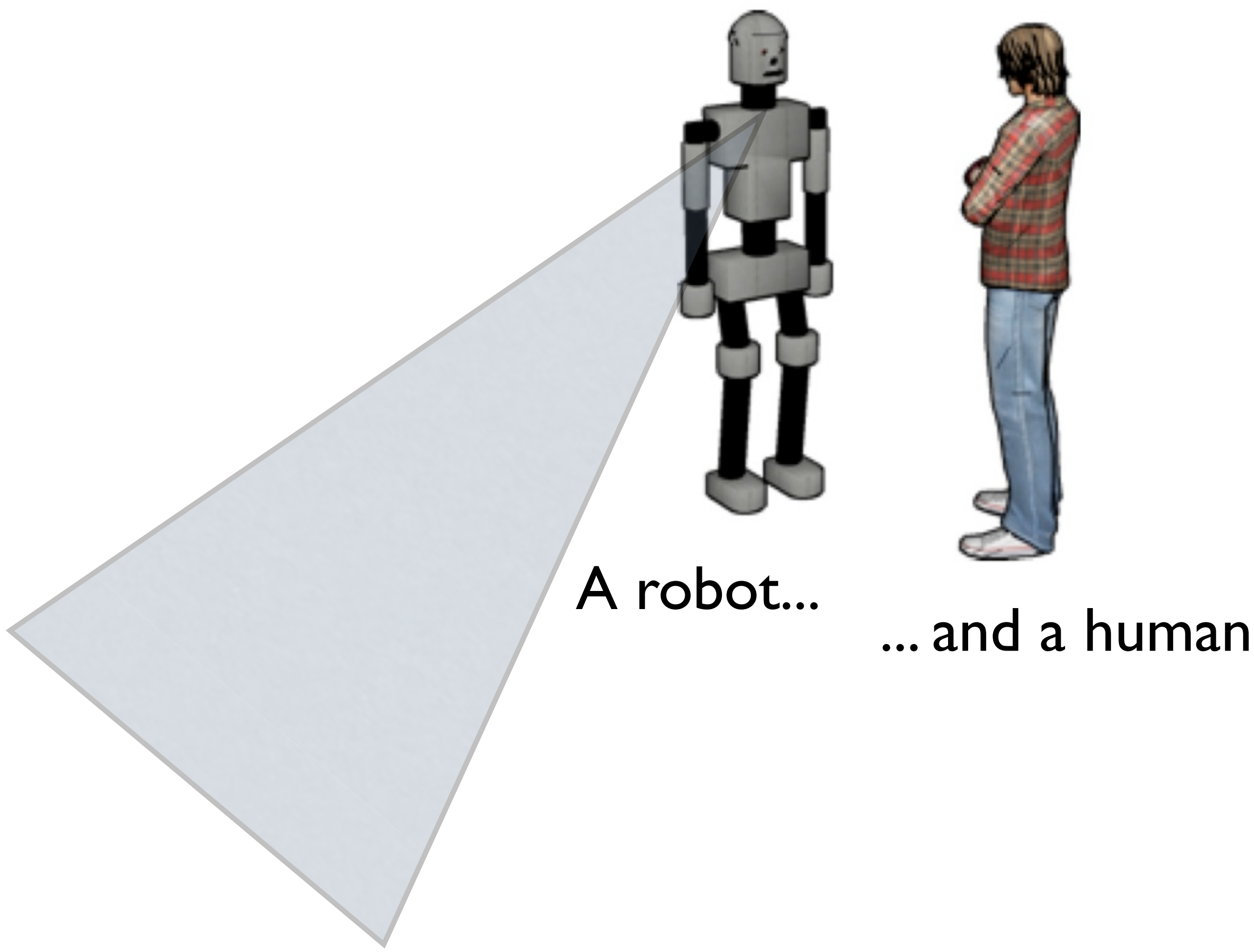
Dialogue management in HRI



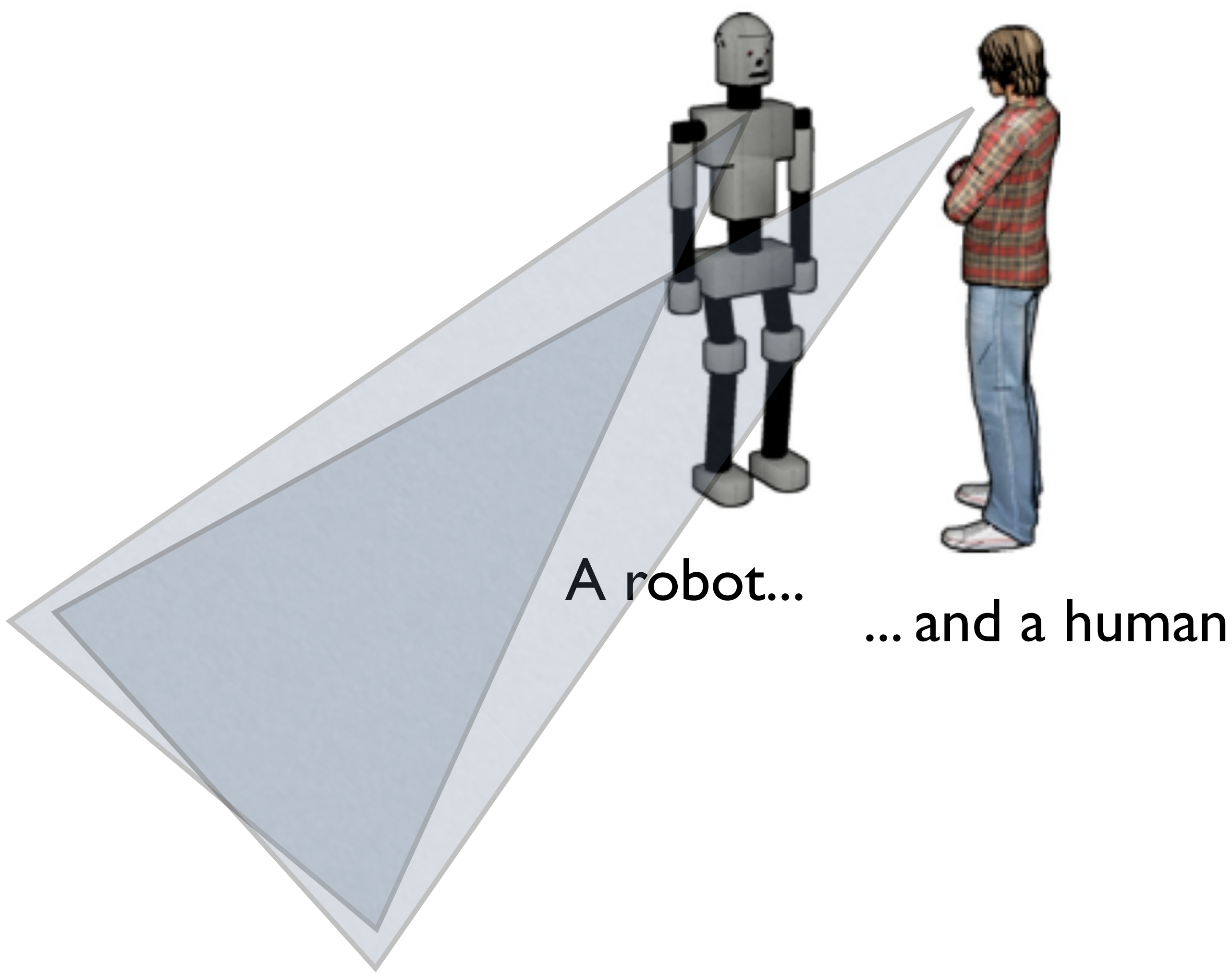
A robot...

... and a human

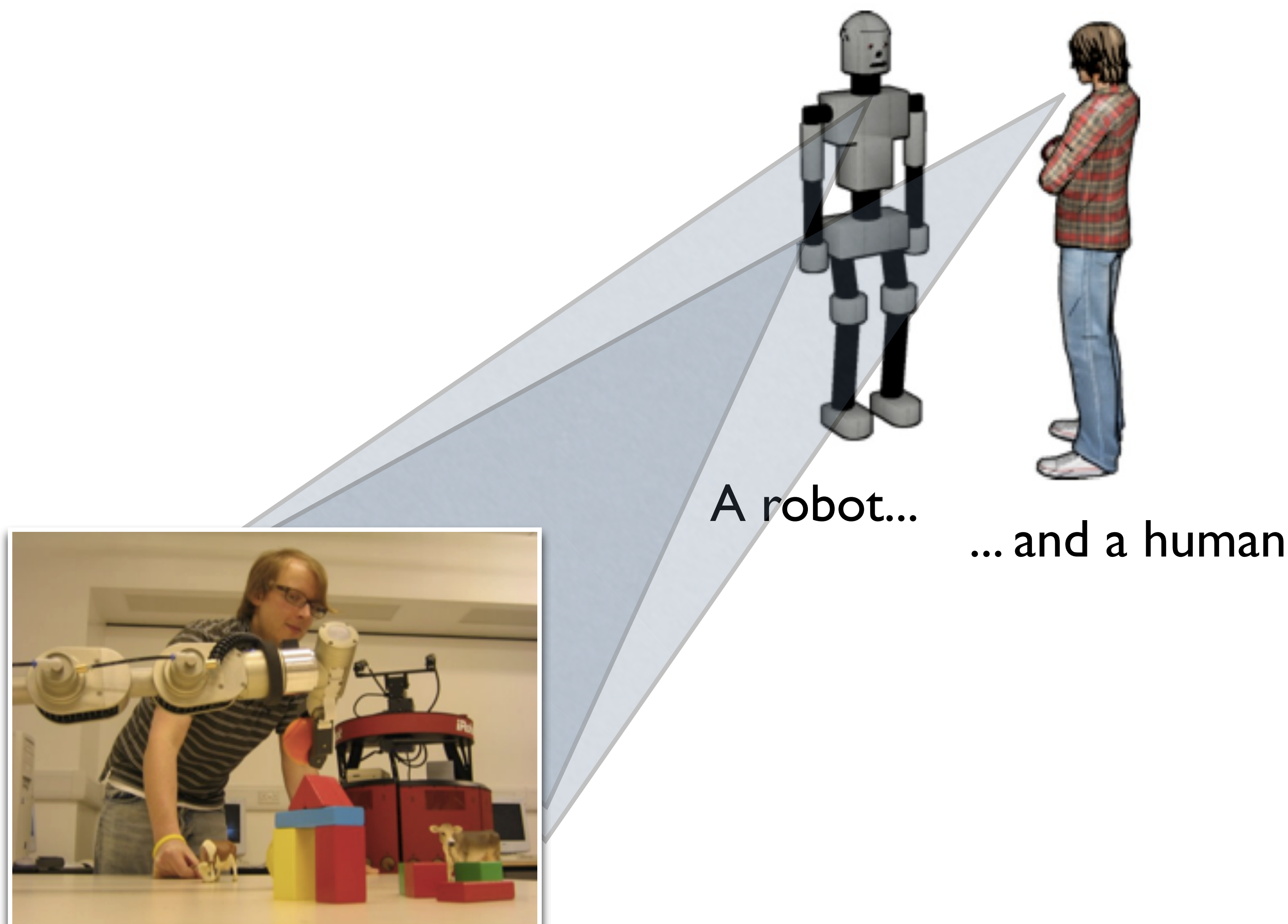
Dialogue management in HRI



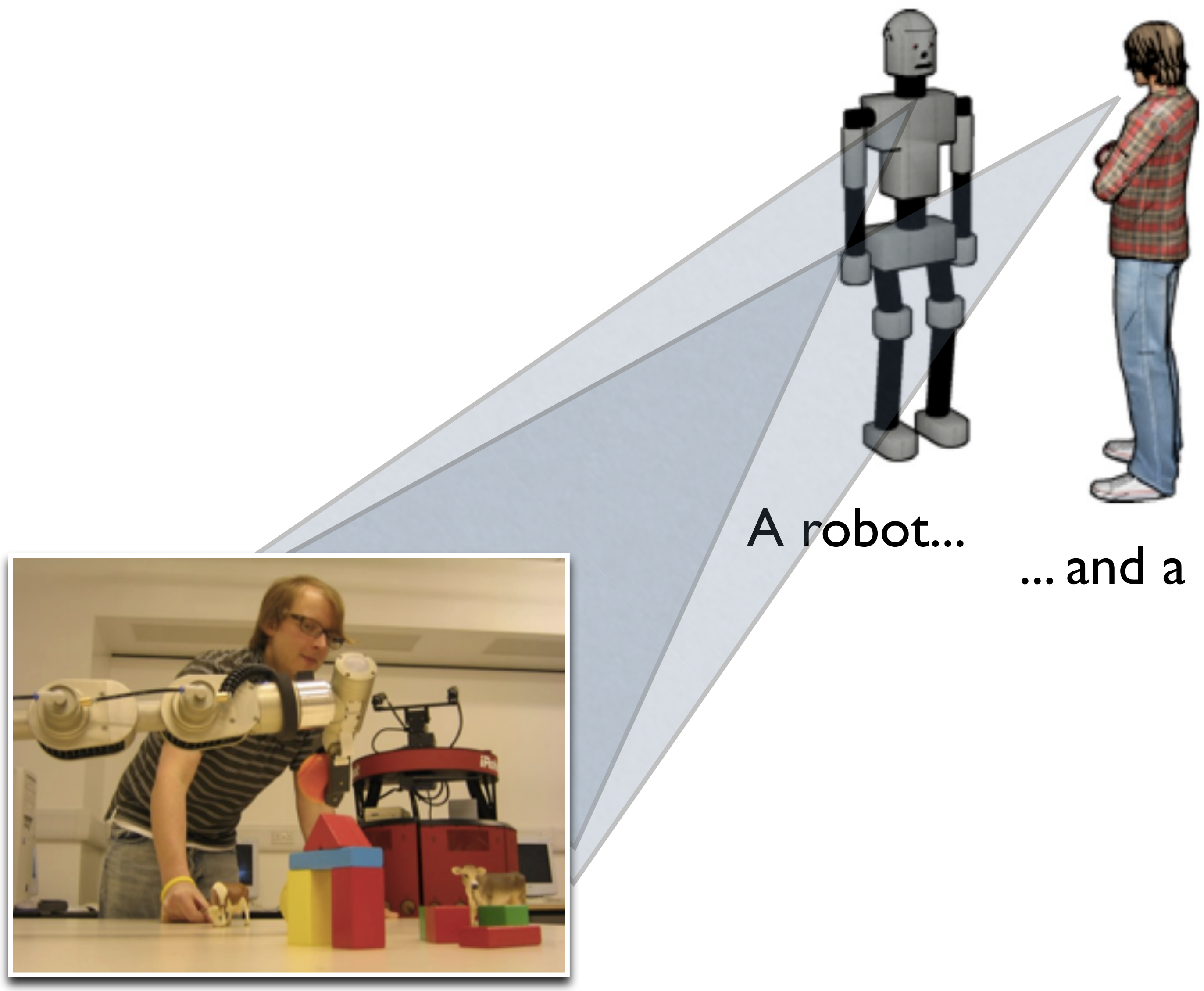
Dialogue management in HRI



Dialogue management in HRI



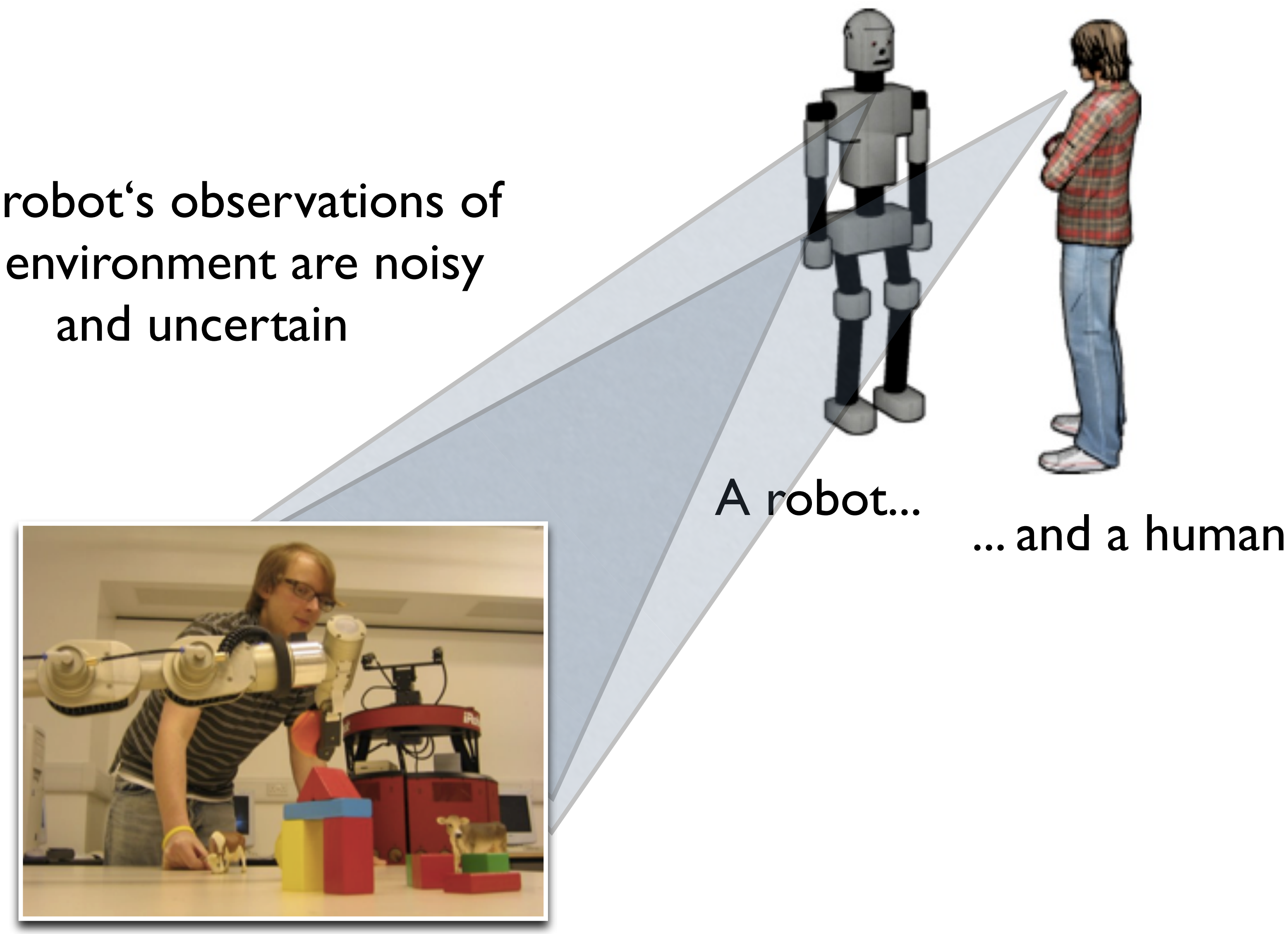
Dialogue management in HRI



in a shared visual scene

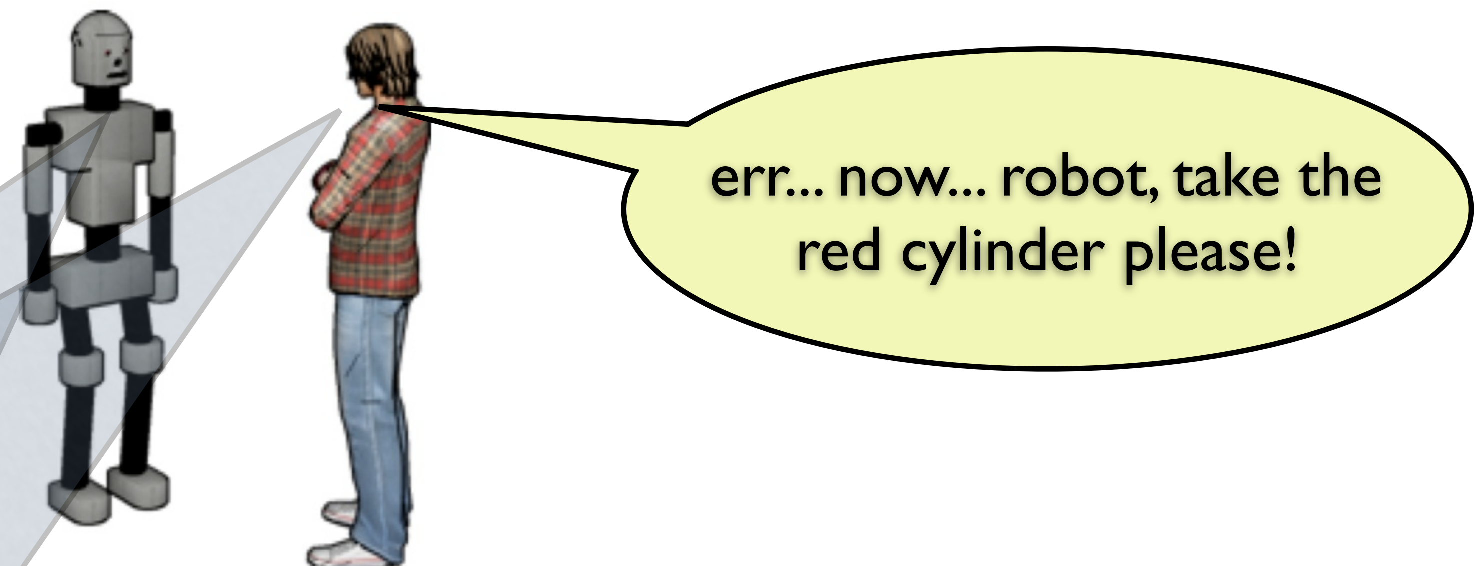
Dialogue management in HRI

The robot‘s observations of the environment are noisy and uncertain



in a shared visual scene

Dialogue management in HRI



The robot‘s observations of the environment are noisy and uncertain

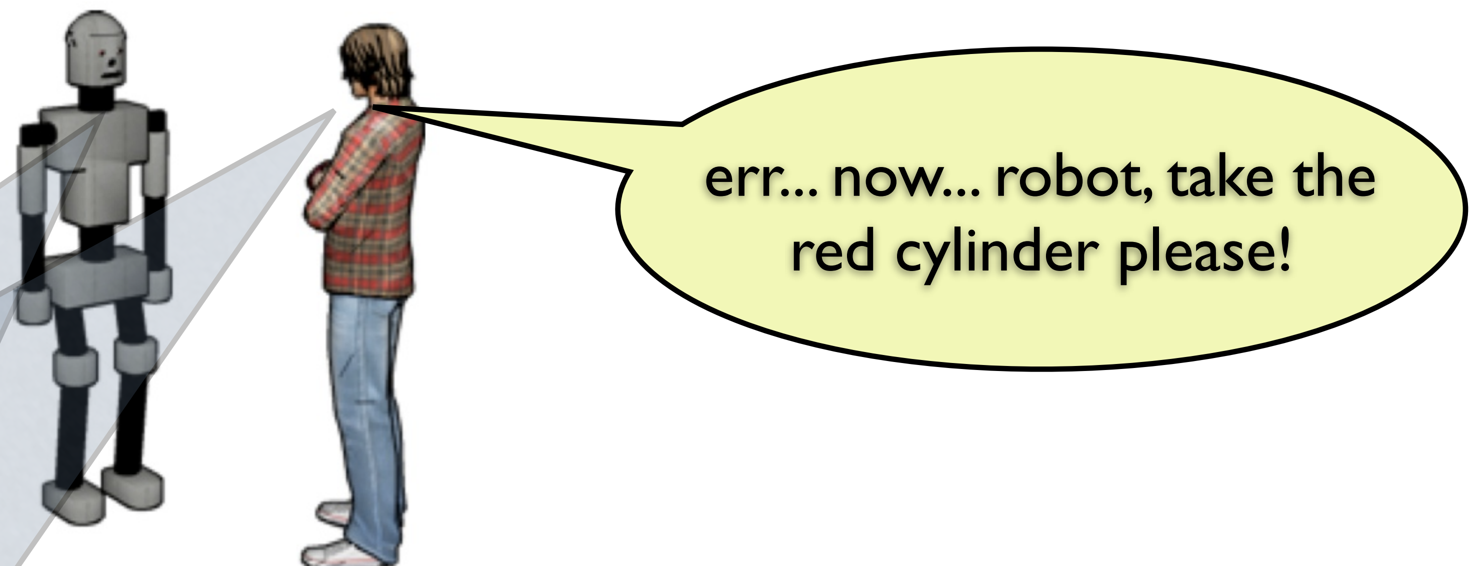
A robot...

... and a human

in a shared visual scene

Dialogue management in HRI

He detects that the human is speaking, and seeks to decode the utterance



The robot‘s observations of the environment are noisy and uncertain

A robot...

... and a human

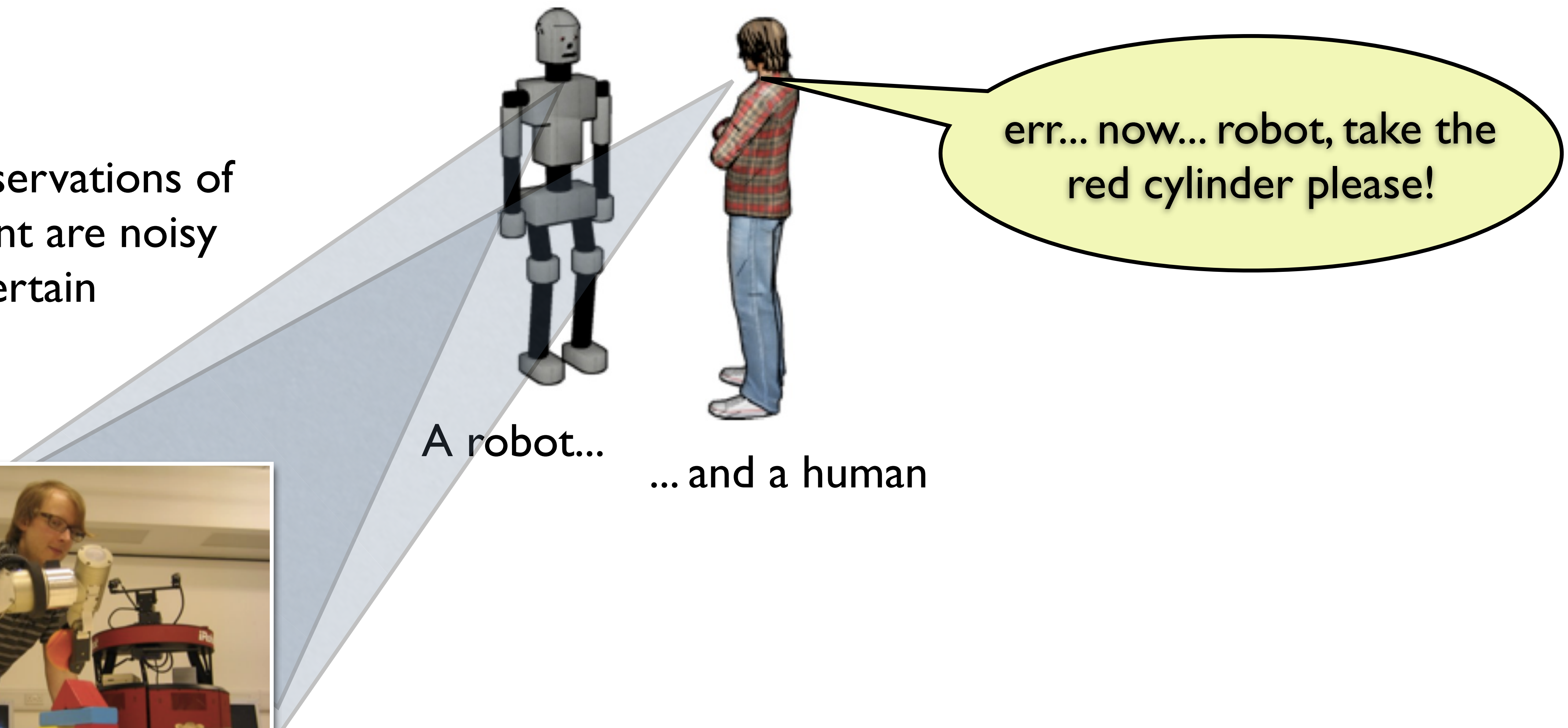
in a shared visual scene

Dialogue management in HRI

He detects that the human is speaking, and seeks to decode the utterance

(which is also noisy and uncertain)

The robot's observations of the environment are noisy and uncertain



A robot...

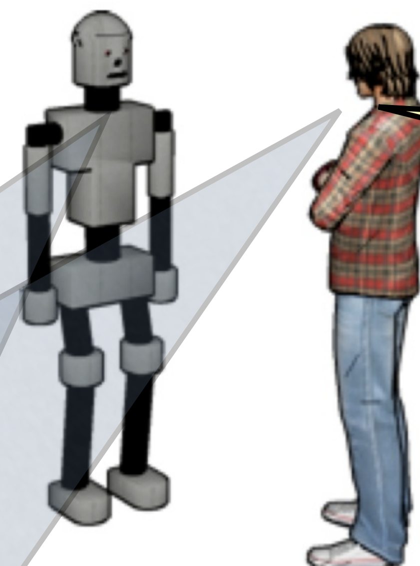
... and a human

in a shared visual scene

Dialogue management in HRI

He detects that the human is speaking, and seeks to decode the utterance

(which is also noisy and uncertain)



err... now... robot, take the red cylinder please!

The robot's observations of the environment are noisy and uncertain

A robot...

... and a human

in a shared visual scene

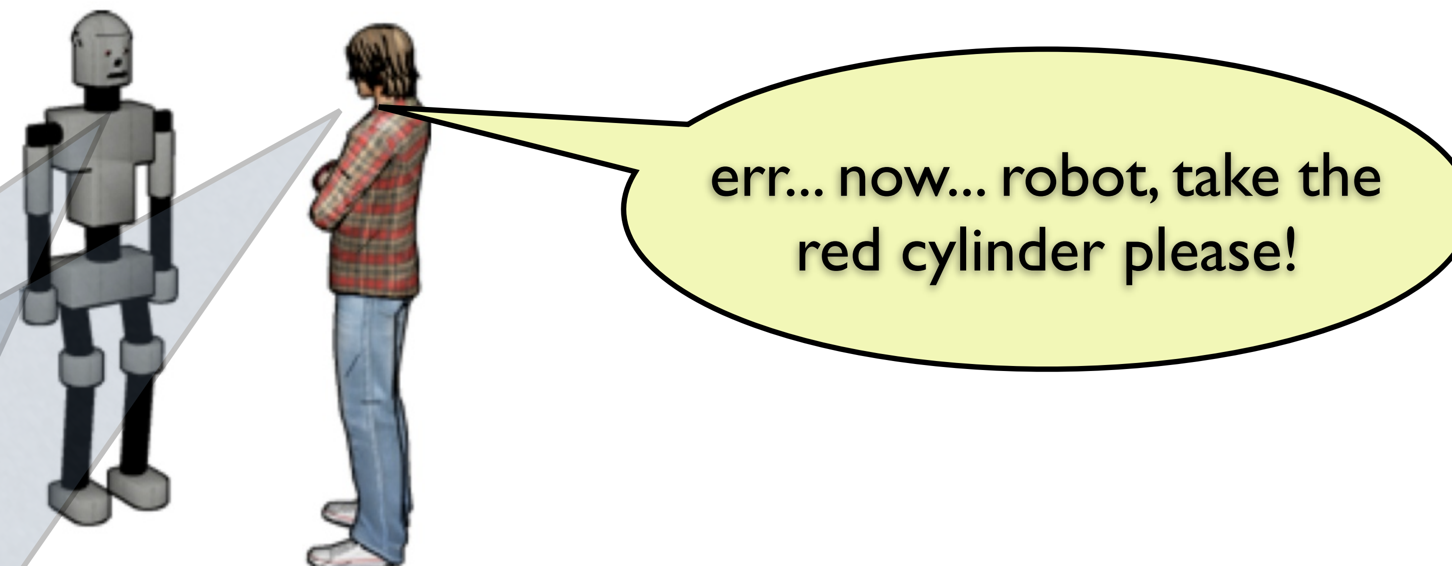
Dialogue management in HRI

The goal is to find the intention behind the utterance

He detects that the human is speaking, and seeks to decode the utterance

(which is also noisy and uncertain)

The robot‘s observations of the environment are noisy and uncertain



A robot...

... and a human

in a shared visual scene

Dialogue management in HRI

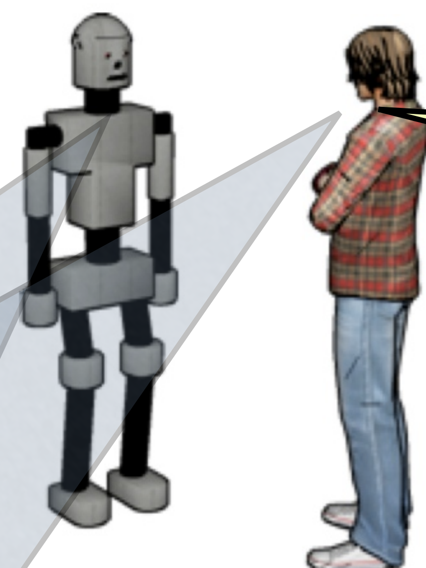
The goal is to find the intention behind the utterance

(including references to the external environment)

He detects that the human is speaking, and seeks to decode the utterance

(which is also noisy and uncertain)

The robot's observations of the environment are noisy and uncertain



err... now... robot, take the red cylinder please!

A robot...

... and a human

in a shared visual scene

Dialogue management in HRI

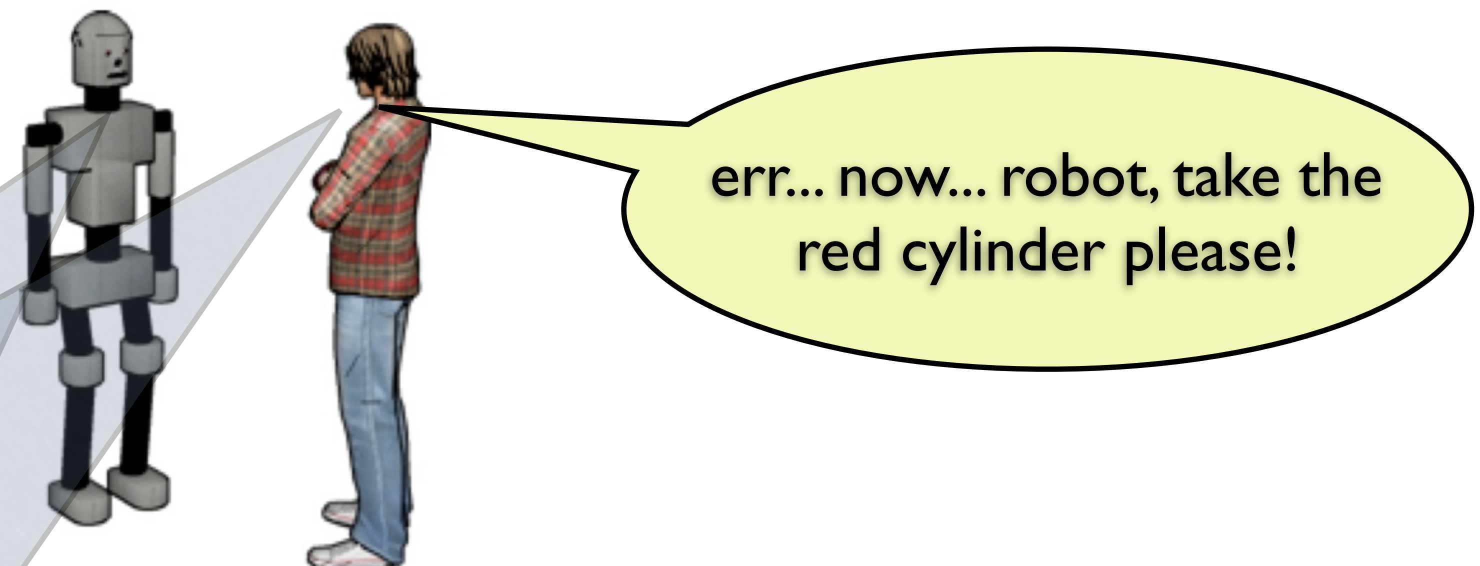
The goal is to find the intention behind the utterance

(including references to the external environment)

He detects that the human is speaking, and seeks to decode the utterance

(which is also noisy and uncertain)

The robot's observations of the environment are noisy and uncertain

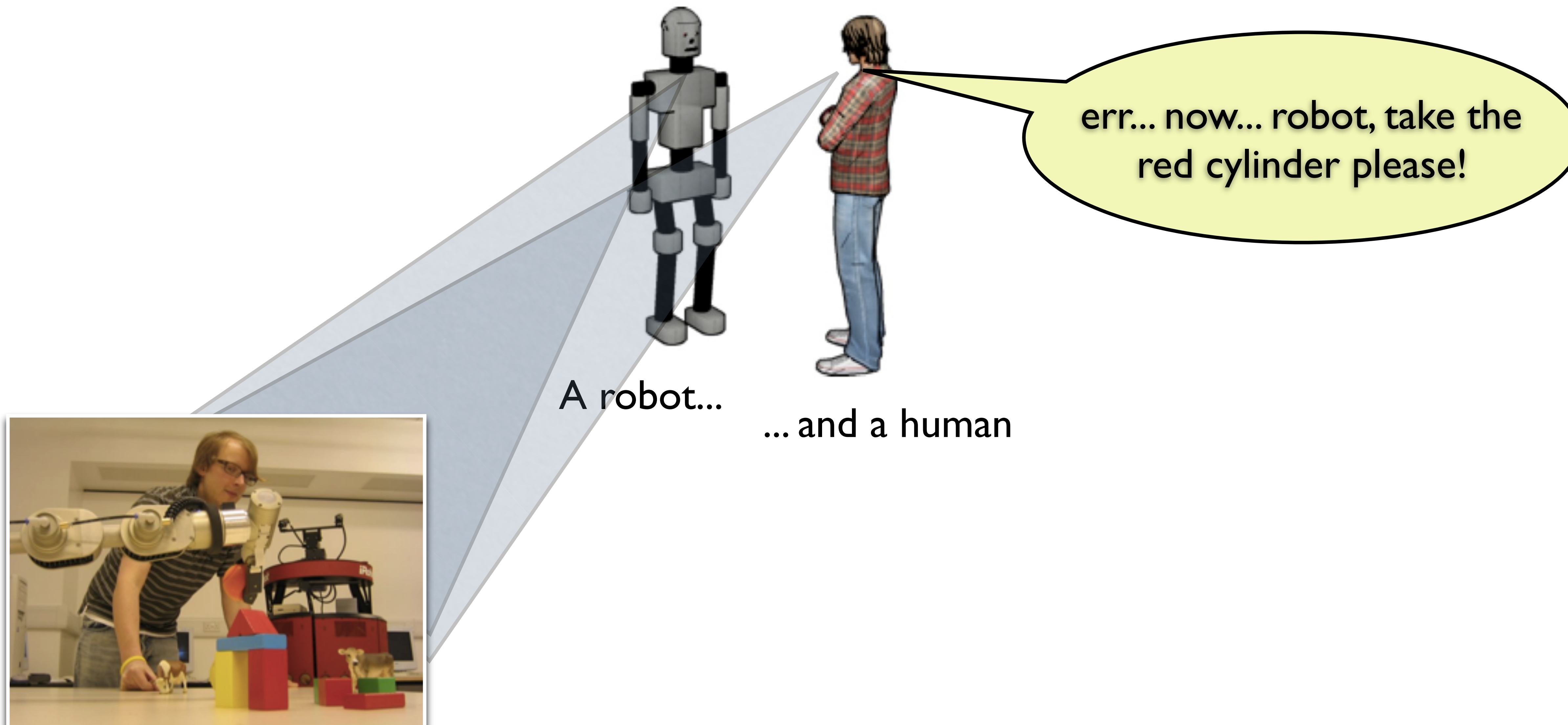


A robot...

... and a human

in a shared visual scene

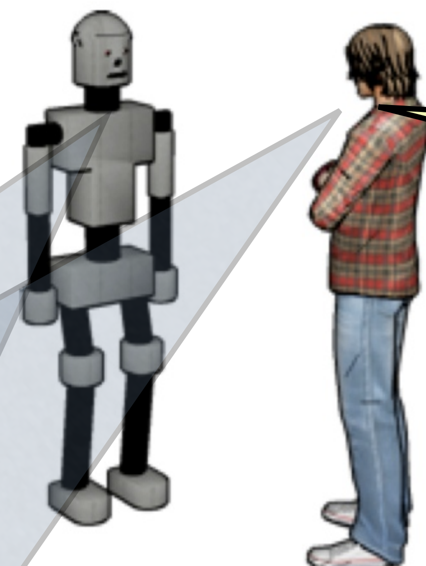
Dialogue management in HRI



in a shared visual scene

Dialogue management in HRI

Once the most probable intention has been computed, the robot must *respond* to it



err... now... robot, take the red cylinder please!

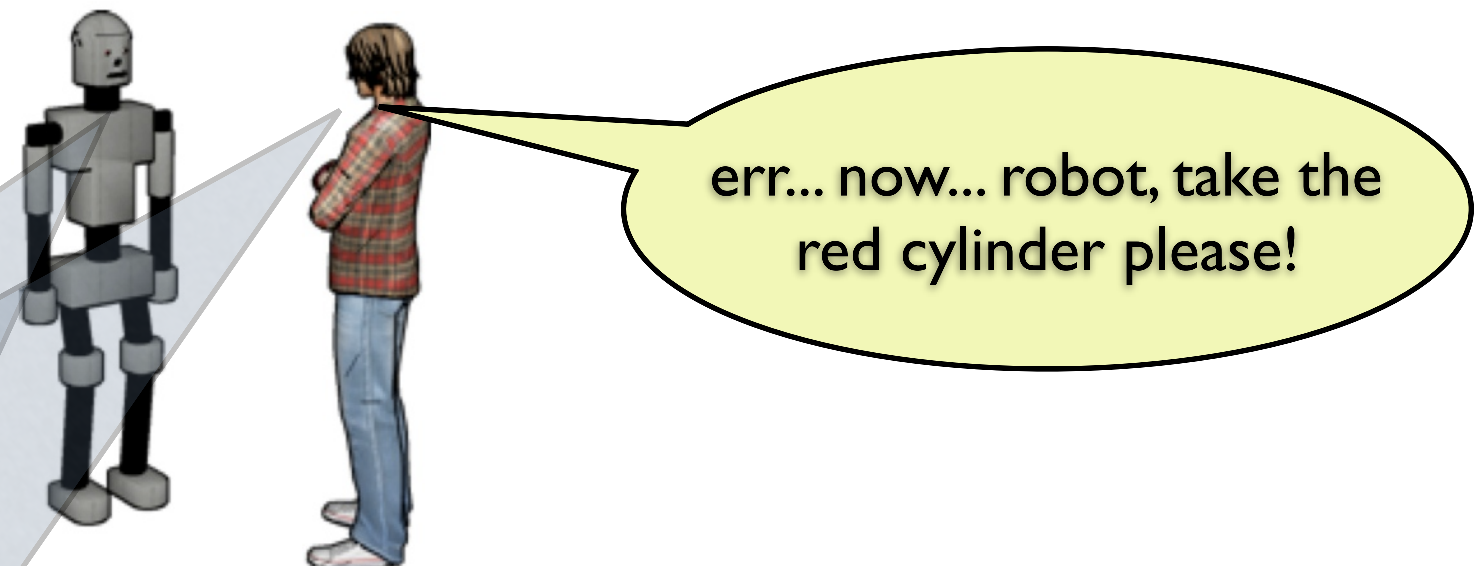
A robot...

... and a human

in a shared visual scene

Dialogue management in HRI

Once the most probable intention has been computed, the robot must *respond* to it



A robot...

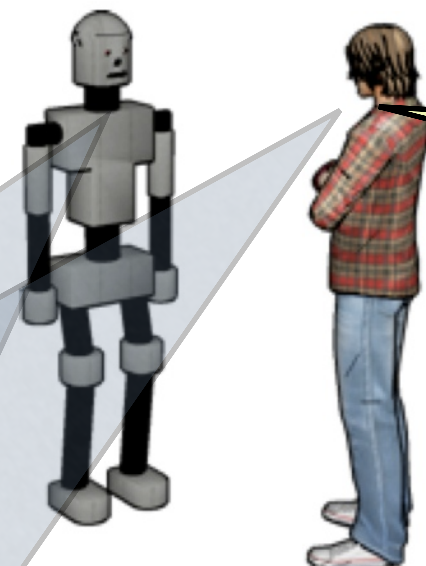
... and a human

in a shared visual scene

Dialogue management in HRI

It searches for the best action(s) to perform at this point

Once the most probable intention has been computed, the robot must *respond* to it



err... now... robot, take the red cylinder please!

A robot...

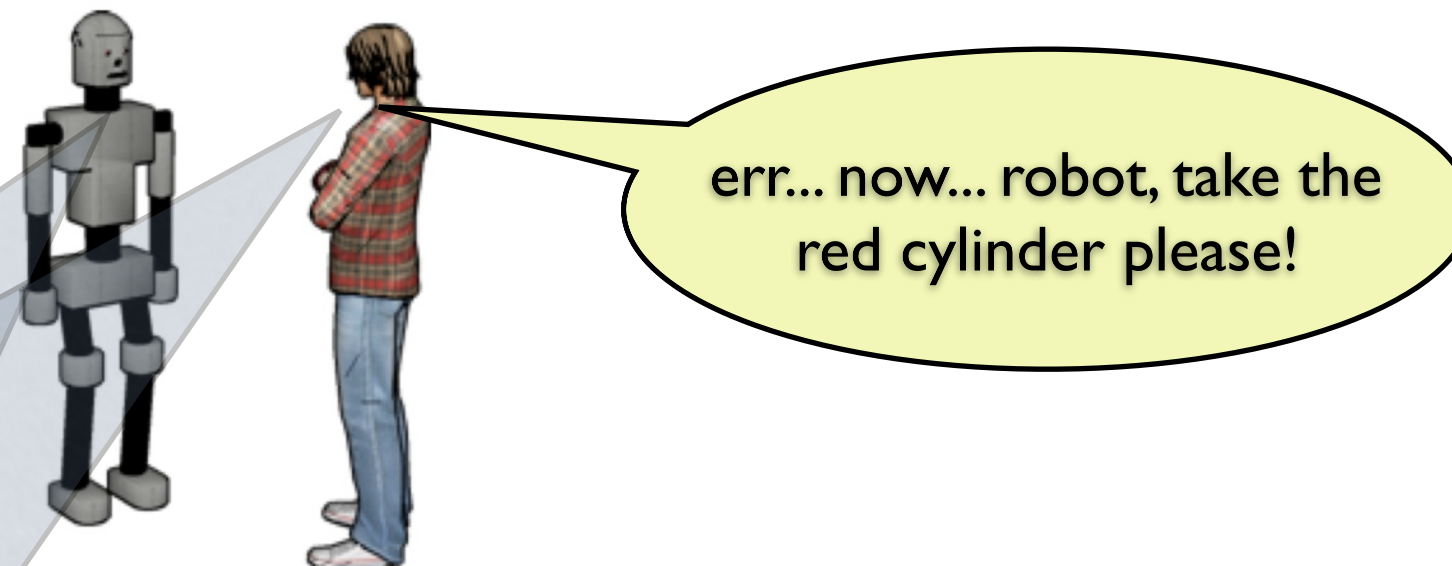
... and a human

in a shared visual scene

Dialogue management in HRI

It searches for the best action(s) to perform at this point

Once the most probable intention has been computed, the robot must *respond* to it



A robot...

... and a human

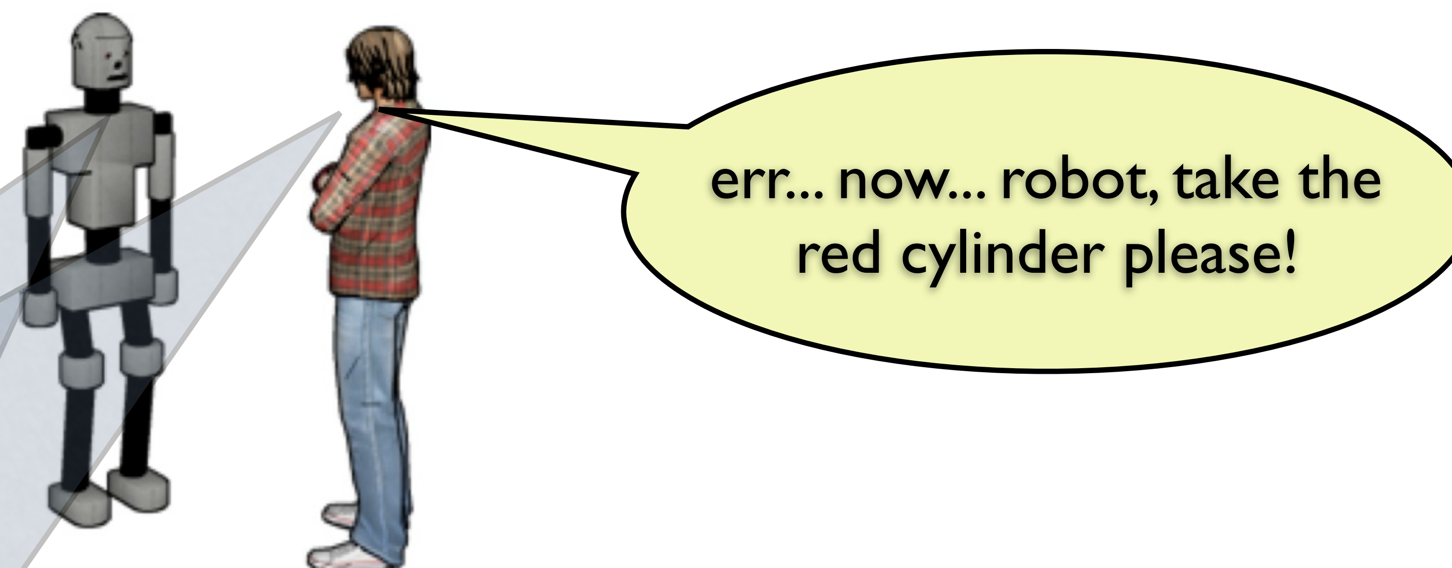
in a shared visual scene

Dialogue management in HRI

It searches for the best action(s) to perform at this point

For instance, a *communicative action*, which generates an utterance

Once the most probable intention has been computed, the robot must *respond* to it



A robot...

... and a human

in a shared visual scene

Dialogue management in HRI

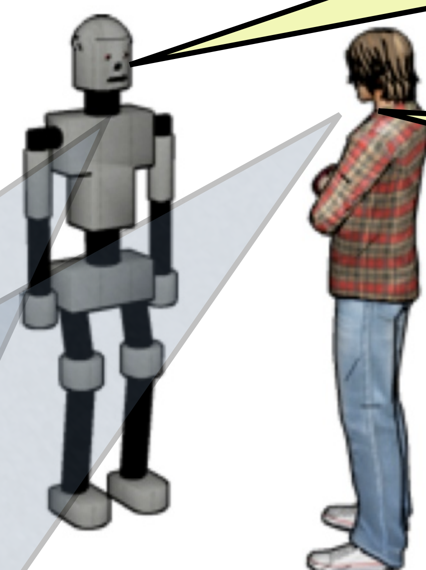
It searches for the best action(s) to perform at this point

For instance, a *communicative action*, which generates an utterance

Yes sure, will do!

Once the most probable intention has been computed, the robot must *respond* to it

err... now... robot, take the red cylinder please!



A robot...

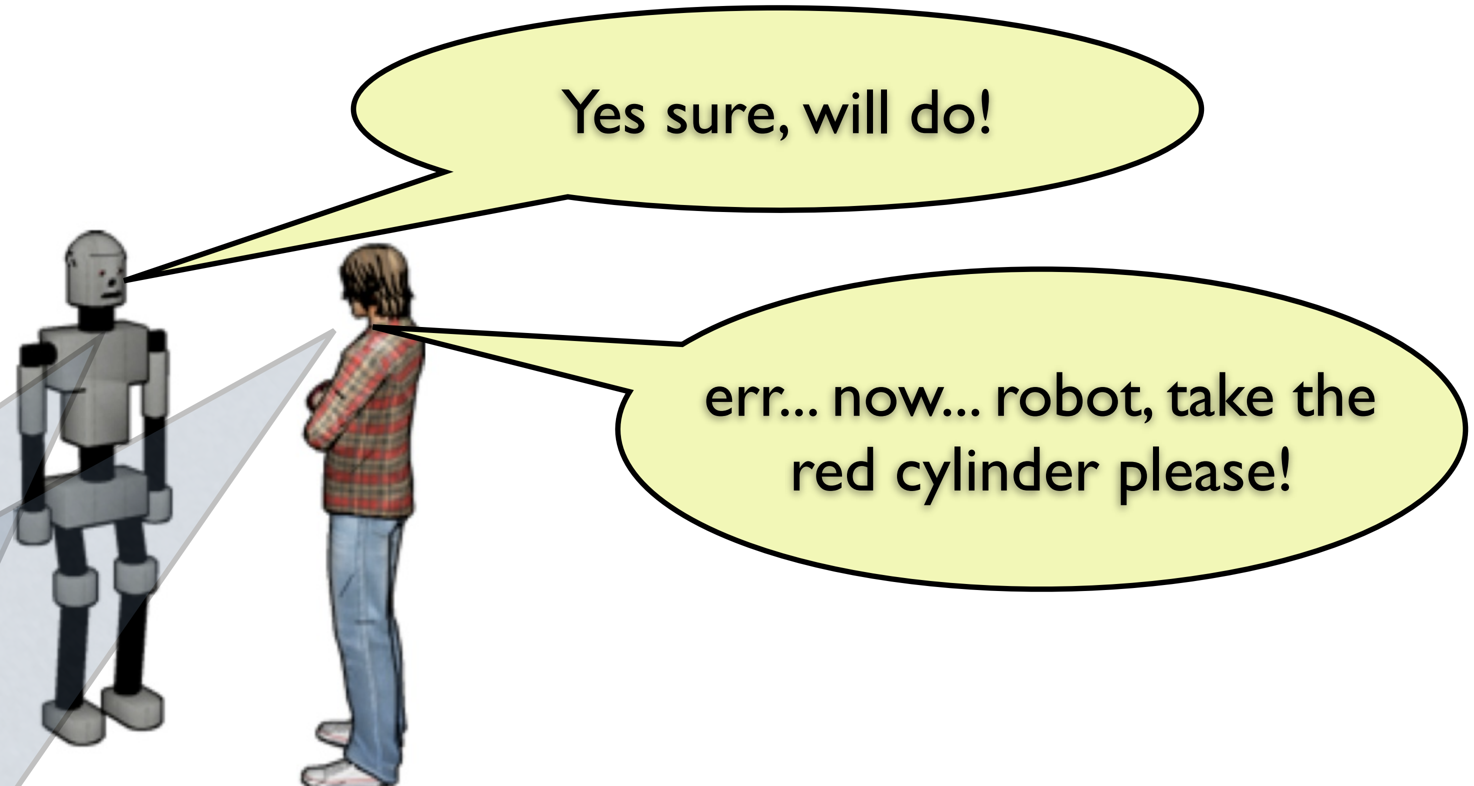
... and a human

in a shared visual scene

Dialogue management in HRI

It searches for the best action(s) to perform at this point

For instance, a *communicative action*, which generates an utterance



Once the most probable intention has been computed, the robot must *respond* to it

A robot...

... and a human

in a shared visual scene

Or a *physical action* which changes the environment

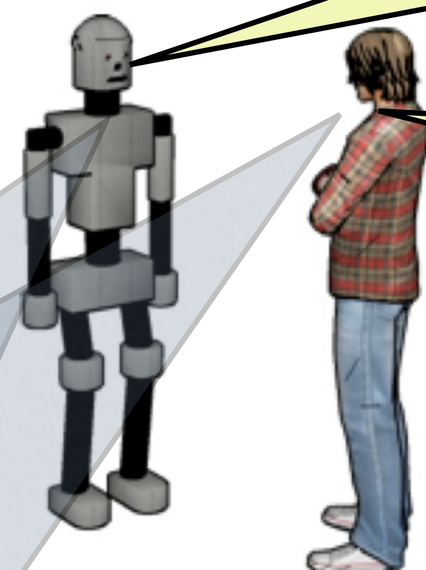
Dialogue management in HRI

It searches for the best action(s) to perform at this point

For instance, a *communicative action*, which generates an utterance

Yes sure, will do!

Once the most probable intention has been computed, the robot must *respond* to it



err... now... robot, take the red cylinder please!

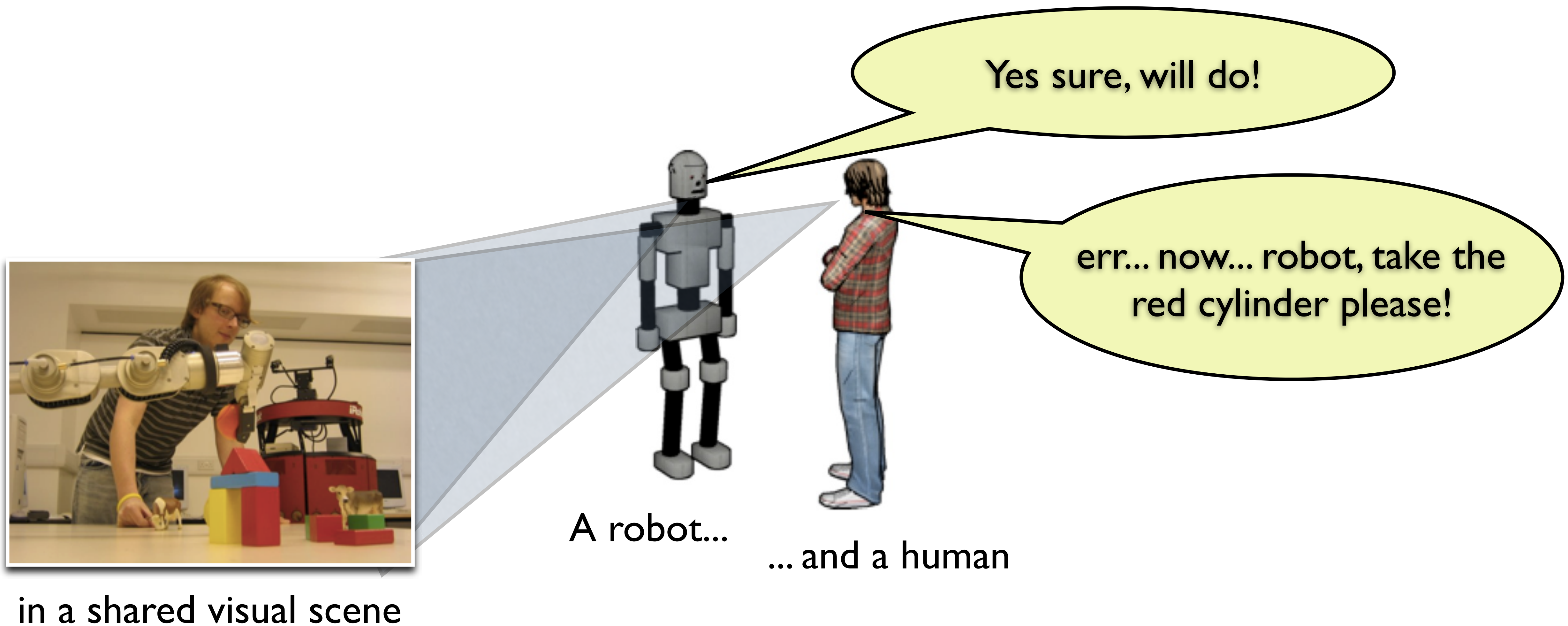
A robot...

... and a human

in a shared visual scene

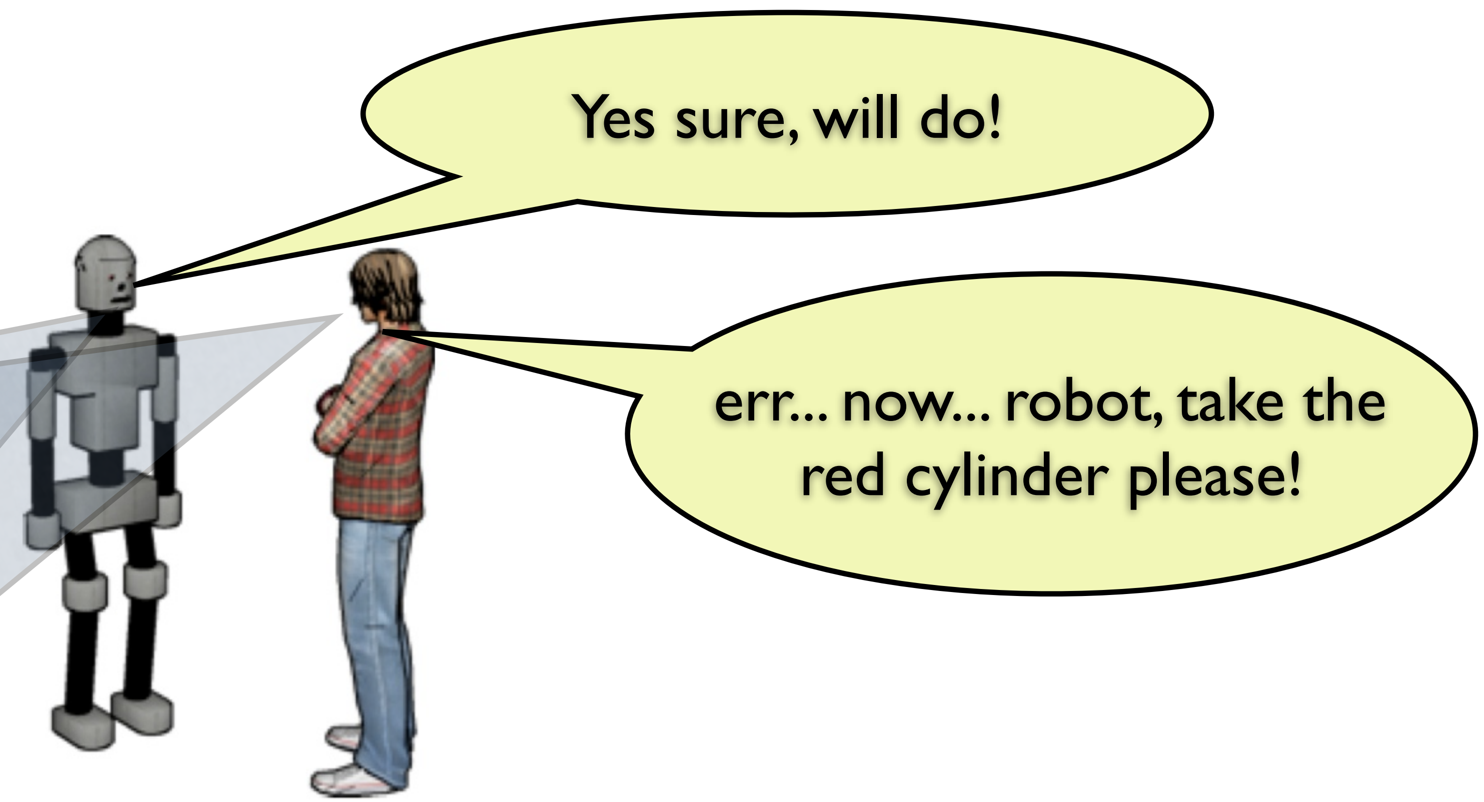
Or a *physical action* which changes the environment

Dialogue management in HRI



Dialogue management in HRI

The robot and the human are both involved in a *collaborative activity*



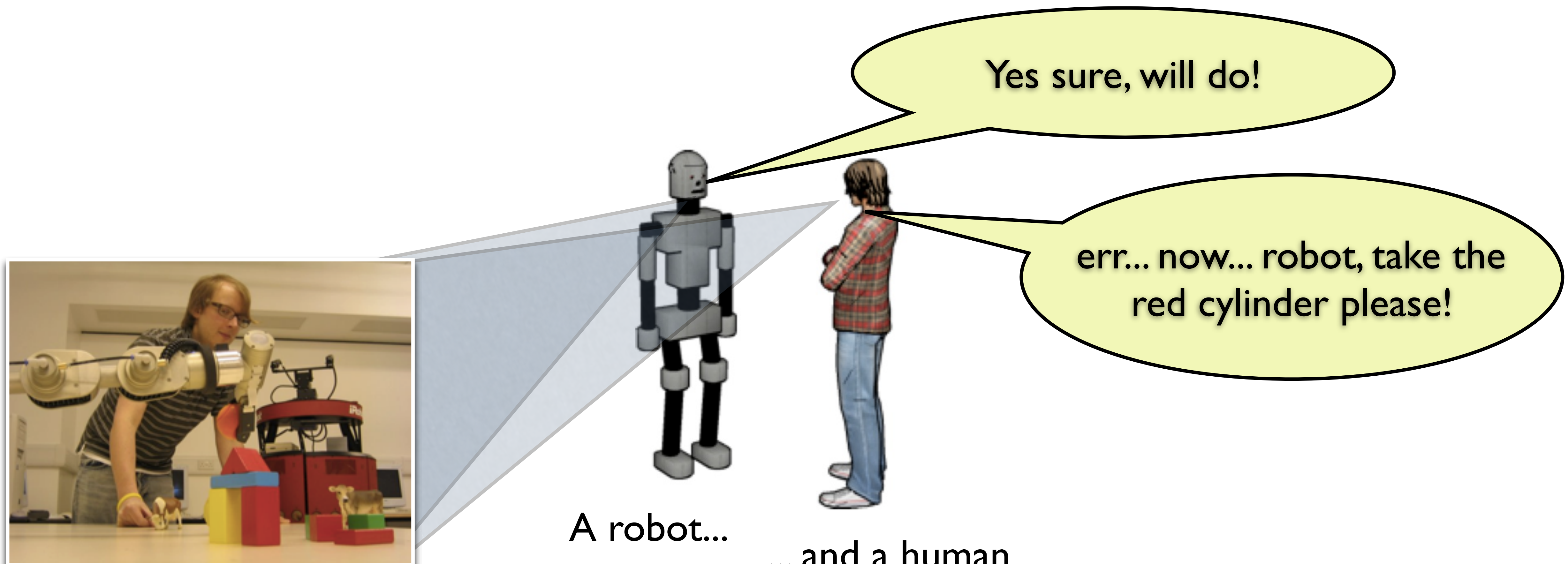
A robot...

... and a human

in a shared visual scene

Dialogue management in HRI

The robot and the human are both involved in a *collaborative activity*

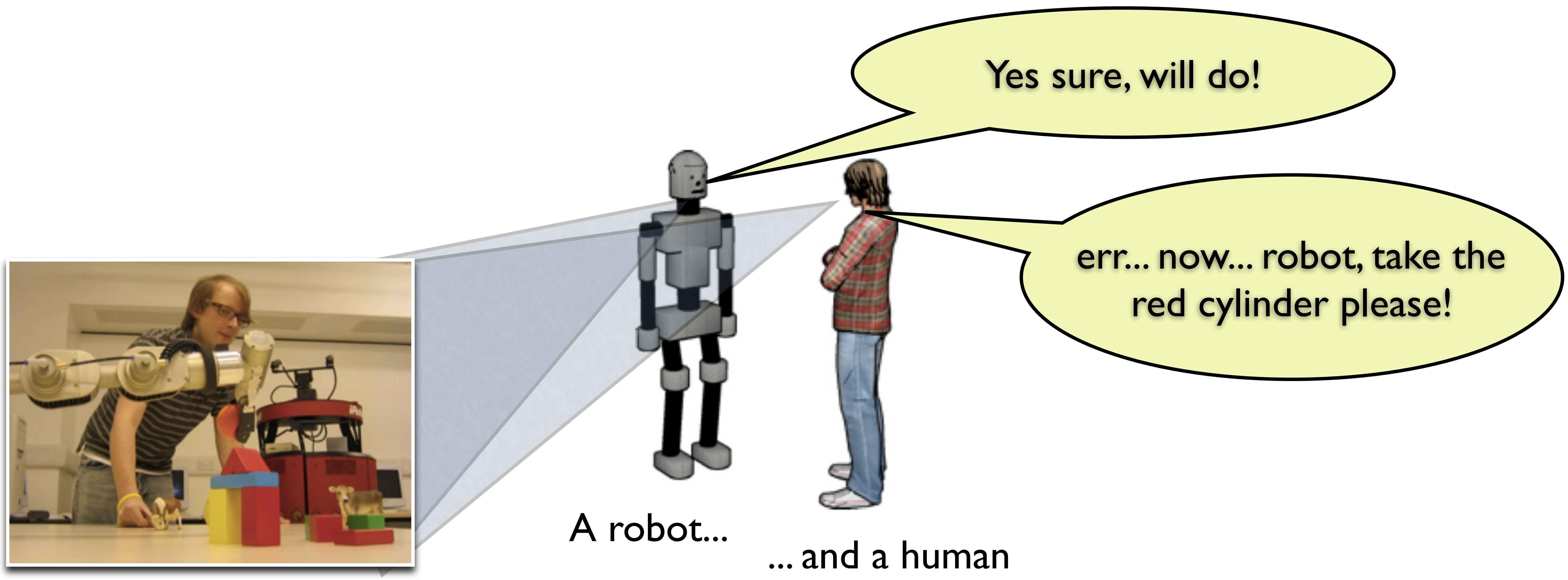


in a shared visual scene

Dialogue management in HRI

The robot and the human are both involved in a *collaborative activity*

The dialogue is part of this activity

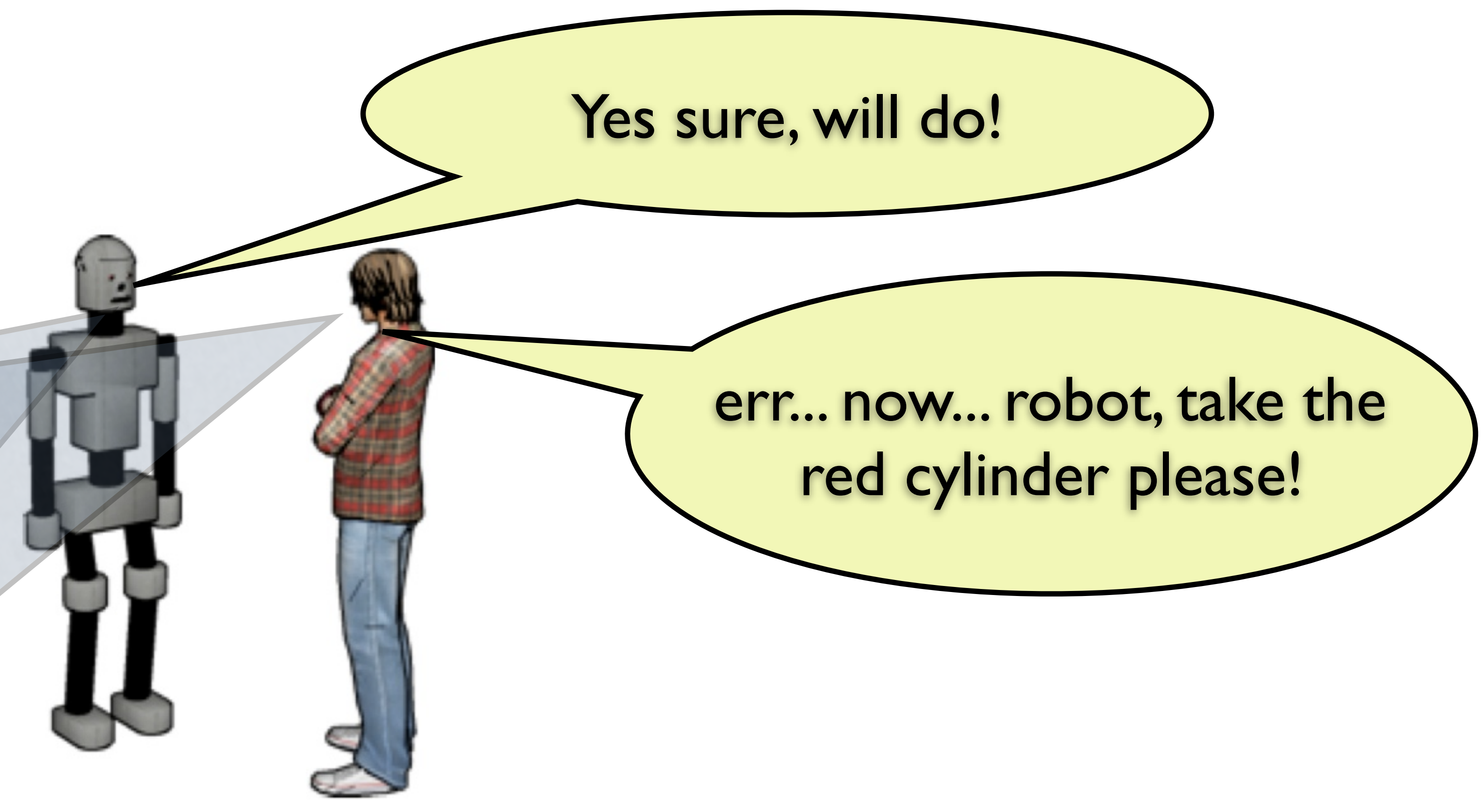


in a shared visual scene

Dialogue management in HRI

The robot and the human are both involved in a *collaborative activity*

The dialogue is part of this activity



A robot...

... and a human

in a shared visual scene

Dialogue management in HRI

The robot and the human are both involved in a *collaborative activity*

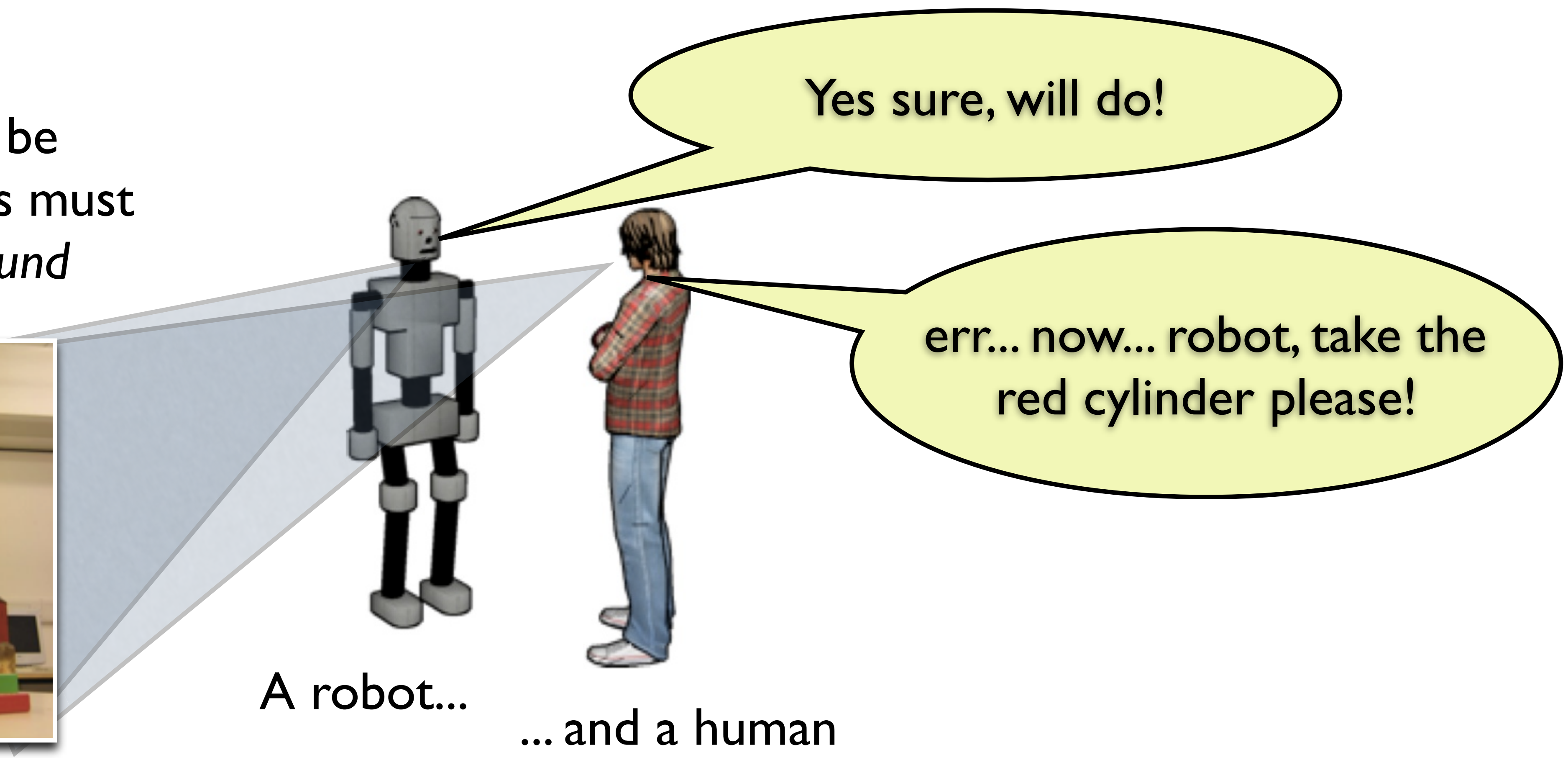
↓

The dialogue is part of this activity

↓

For the interaction to be successful, the participants must establish a *common ground*

in a shared visual scene

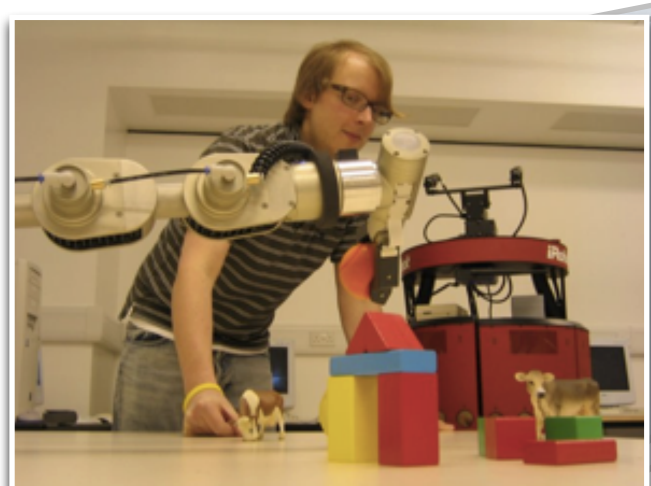


Dialogue management in HRI

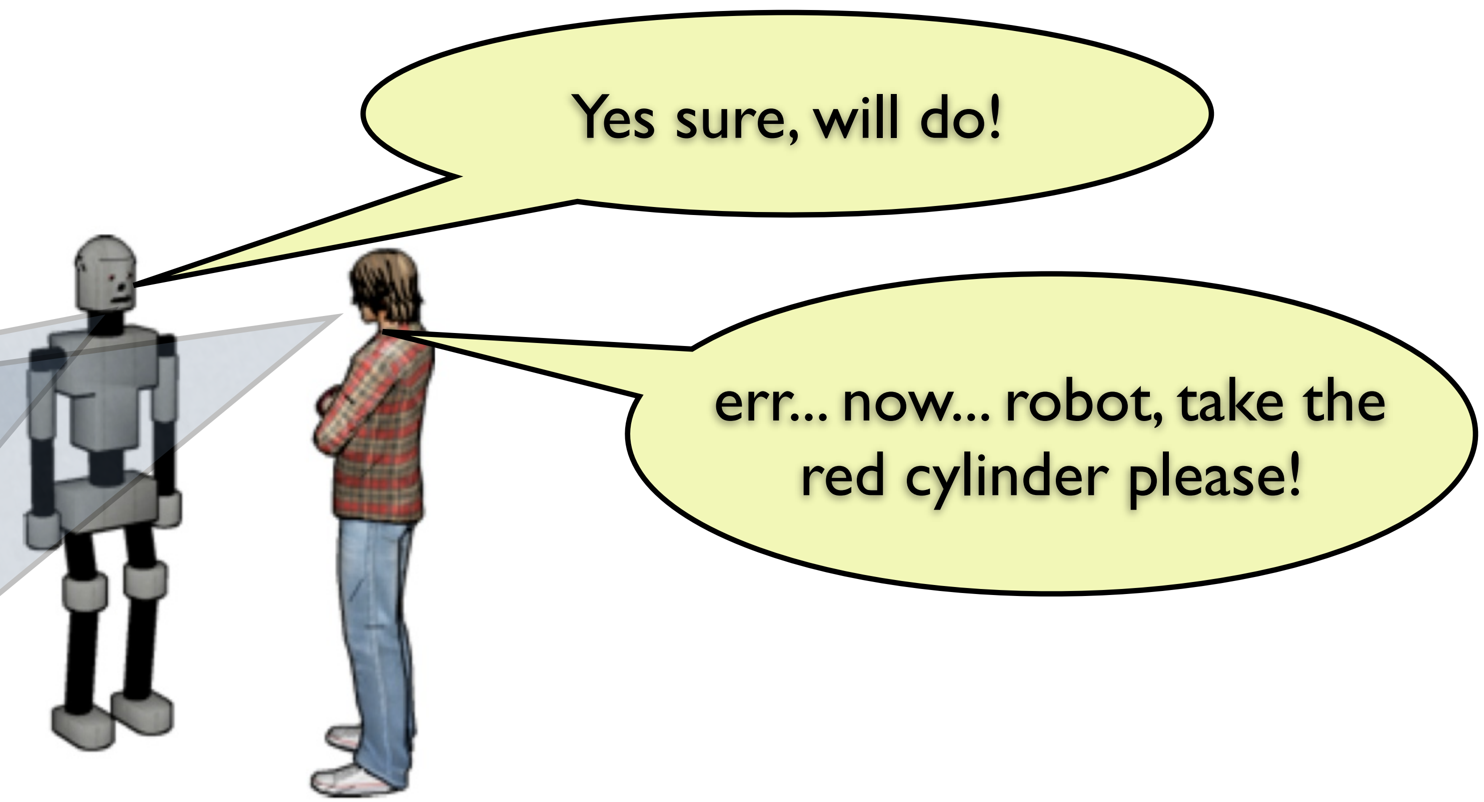
The robot and the human are both involved in a *collaborative activity*

The dialogue is part of this activity

For the interaction to be successful, the participants must establish a *common ground*



in a shared visual scene



A robot...

... and a human

Dialogue management in HRI

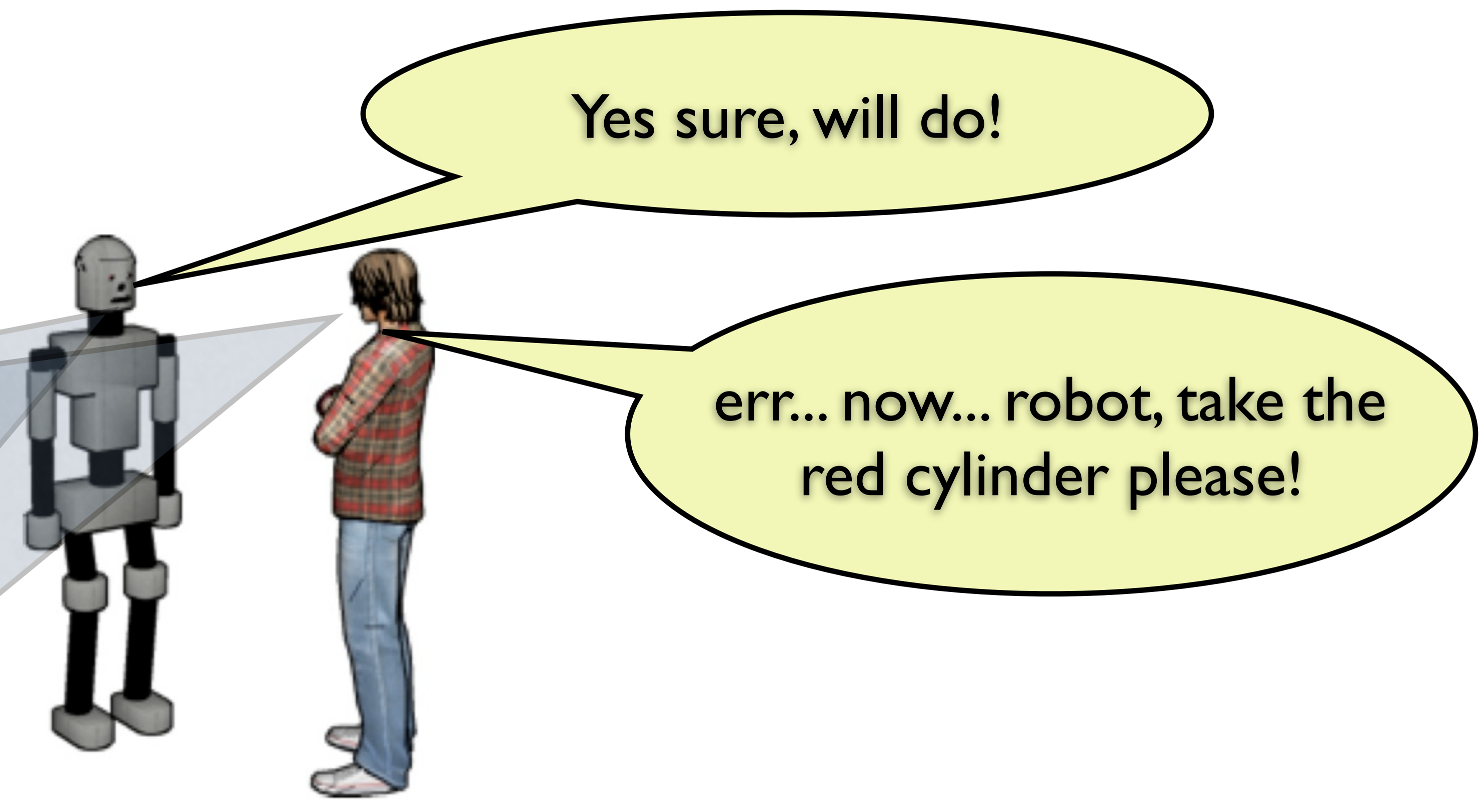
The robot and the human are both involved in a *collaborative activity*

The dialogue is part of this activity

For the interaction to be successful, the participants must establish a *common ground*

Overall, the robot's actions must reflect:
- the goals of the activity (what is to be done),
- its history (what has been done)
- the environment state (what's my reality)
- models of the other agents (what's their reality)
- the attentional state (what is the current focus)

in a shared visual scene

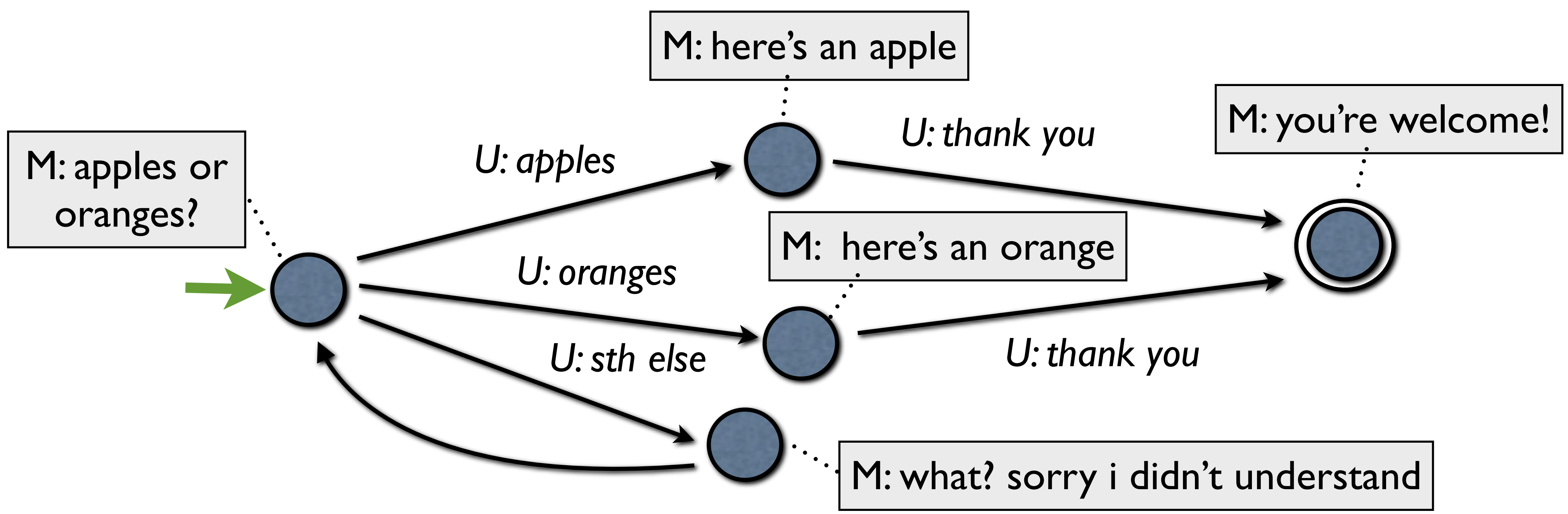


A robot...

... and a human

Approaches to dialogue modelling

- **Finite-state automata:**
  - interaction modelled as a *graph*
  - the nodes represent *machine actions*
  - and the edges possible (mutually exclusive) *user responses*



- FSAs are easy to design, but only allow for very *rigid* and *scripted* types of interactions. Not suited for flexible conversations

Approaches to dialogue modelling

- **Information-state approach**
  - The *information state* (IS) is a central repository representing information about the current interaction
  - IS can encode the mental states, beliefs and intentions of the speakers, common ground, dialogue context
  - The observation of new dialogue moves triggers information state *updates* (specified in ad-hoc rules)
  - *Action selection* is then performed on the updated IS
- More generic & flexible approach to dialogue management, but can be hard to design

[David Traum and Staffan Larsson 2003]

Approaches to dialogue modelling

- **Plan-based approaches**
  - Dialogue management viewed as *planning* problem
  - Dialogue interpretation as *plan recognition*
  - Similarly to IS, plan-based approaches typically rely on rich, logic-based models of the task domain
  - Generally require the system designer to spell out a complete, detailed model of the interaction
  - Rarely used beyond academic prototypes

[Thomason, R. H., & Stone, M. 2006]

Shortcomings of handcrafted policies

- Unreliable speech recognition
- Rigid, repetitive structure of the interaction
- Irritating confirmations & acknowledgements
- No user or context adaptivity

“Saturday night live” sketch comedy, 2005

Shortcomings of handcrafted policies

- Unreliable speech recognition
- Rigid, repetitive structure of the interaction
- Irritating confirmations & acknowledgements
- No user or context adaptivity

“Saturday night live” sketch comedy, 2005

Outline of the talk

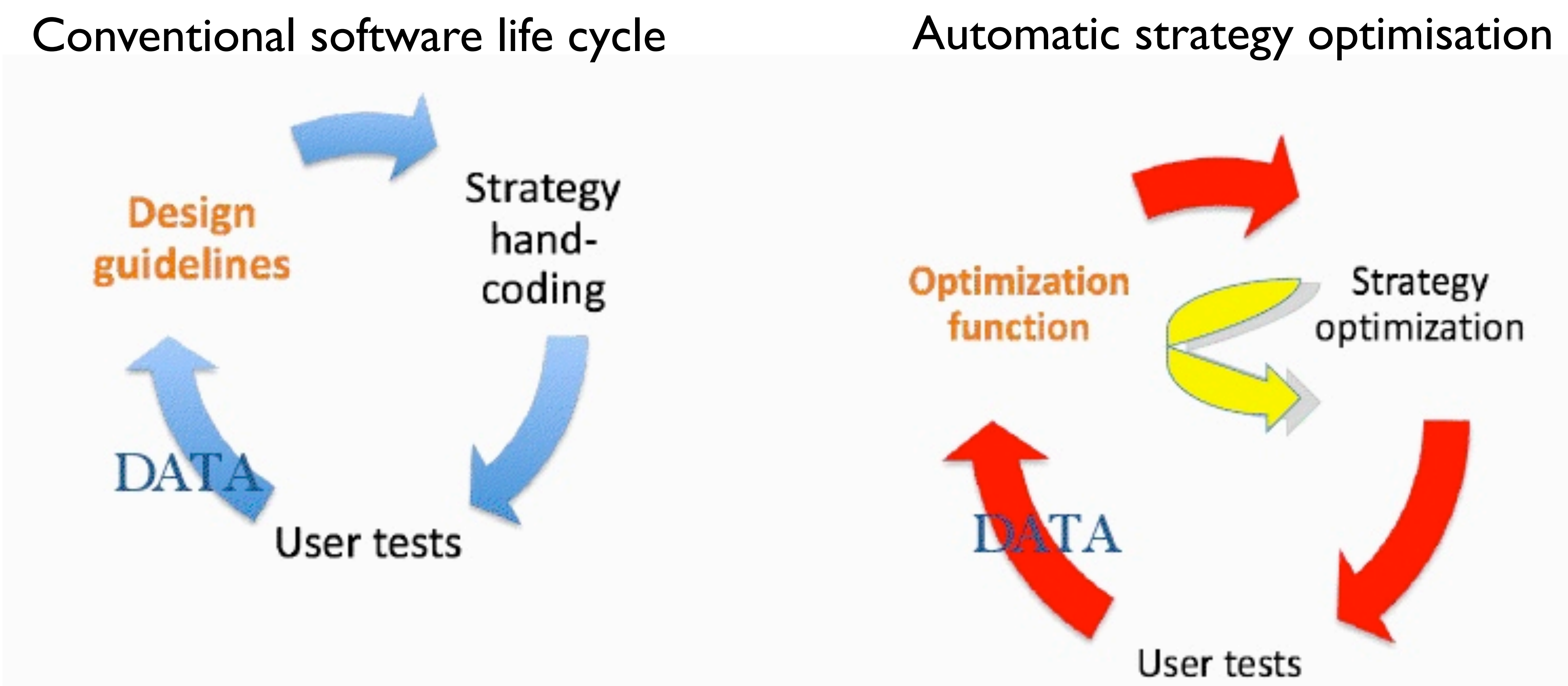
- Background
  - Generalities about dialogue systems
  - Dialogue management, dialogue models
  - **Learning dialogue policies**
- Ongoing work
  - General vision
  - Current research directions
  - Implementation & evaluation
- Conclusion

Approaches to dialogue modelling

- **Machine-learning approaches**
  - Generally model the interaction as a Markov Decision Process (MDP) or as a Partially Observable Markov Decision Process (POMDP)
  - The dialogue policies are automatically *learned from experience* (by reinforcement learning) instead of being handcrafted by the system designer
  - More principled account of uncertainty
  - Leads to more natural and less rigid interaction styles
  - Getting training data & user simulators can be difficult

[Lemon, O., & Pietquin, O. 2007]
[Williams, J. D., & Young, S. 2007]

Design/optimisation of policies



Design by `Best practices'

(Paek 2007)

Automatic design by optimization function

(= "programming by reward")

(Drawings borrowed from slides made by O. Lemon)

Why optimise dialogue policies?

- Avoids handcrafting
- Data-driven development cycle
- Provably optimal policies given a specified objective function (i.e. reward function)
- Precise mathematical model for action selection, with explicit account of uncertainties
- Leads to more flexible and adaptive systems
- Can encode complex trade-offs

Markov Decision Process

- Dialogue can be easily captured as an MDP
- What is a Markov Decision Process? Formally, it is defined as a tuple **<S,A,T,R>**, where:
  - **S** is the *state space* (space of possible, mutually exclusive "situations" in the domain, from the agent's viewpoint)
  - **A** is the *action space* (possible actions for the agent)
  - **T** is the *transition function*, defined as $T(s, a, s') = P(s'|s, a)$. It is the probability of arriving to state **s'** after executing action **a** in state **s**.
  - **R** is the *reward function*, defined as $R : S \times A \rightarrow R$. It is a real number encoding the utility for the agent to perform action **a** while in state **s**.

[Sutton, R. S., & Barto, A. G. 1998]

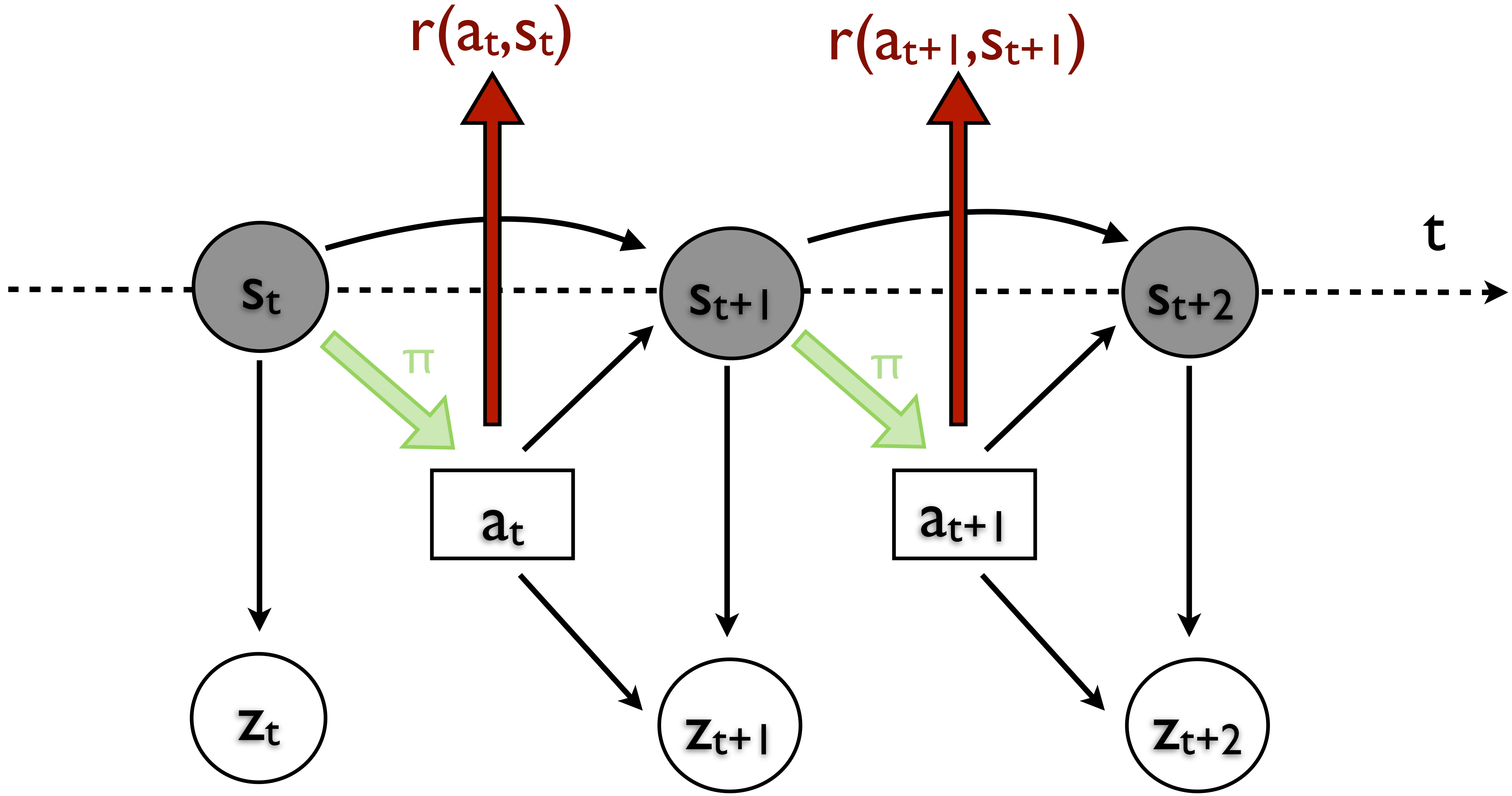
Partially observable MDPs

- MDPs can capture the uncertainty of action *outcomes*, but still assume that the *current* state **s** is known with certainty
- *Partially Observable Markov Decision Process* extend MDPs by incorporating state uncertainty. POMDPs are formally defined as a tuple <S,A,O,T,Ω,R>, where:
  - S,A,T and R are defined similarly to MDPs
  - **O** is the observation space, i.e. the set of input observations which can be captured by the agent
  - **Ω** is the observation function, defined as $\Omega(z,a,s') = P(z|a, s')$. It defines the probability of observing **z** after executing action **a** when the true (hidden) state of the world is **s'**

[Young, S., Gasic, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., & Yu, K. 2010]

POMDPs, graphically

Hidden variables are greyed. Actions are represented as rectangles to stress that they are system actions rather than observed variables. Arcs into circular nodes = influence, whereas arcs into squared nodes = informational.



Only one state is shown here at each time step, but note that the policy π is function of the full belief state rather than a single (unobservable) state.

POMDPs in the dialogue domain

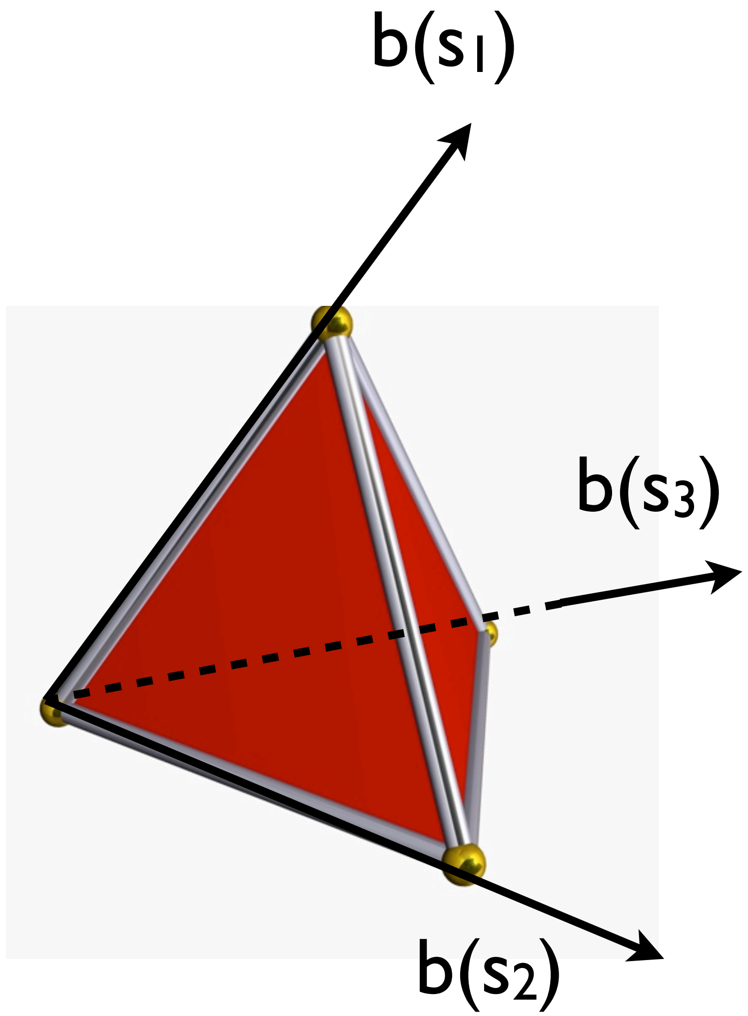
- *State space*: set of possible states of the interaction;
- *Action space*: set of possible dialogue moves;
- *Observation space:* set of possible interpretations of linguistic utterances, together with their confidence score;
- *Transition function*: definition of the dialogue "dynamics"
- *Observation function*: "sensor model" between utterance interpretations and their actual (hidden) intentions
- *Reward function*: big positive reward for long-term goals (e.g. the retrieval of important information), and small negative rewards for various "inconveniences" (e.g. prompting the user to repeat).

The reward function

- The rewards can express:
  - the goals of communication (providing/extracting a given piece of information, performing a joint task, etc.)
  - constraints on communication (e.g. Grice's Maxims)
  - Rewards/punishments can reflect both individual and cooperative goals, linguistic and non-linguistic actions
- An optimised dialogue policy will seek to maximise the cumulative expected reward
- Pivotal role for designing good policies!

Belief state

- A key idea of POMDP is the assumption that the state of the world is not directly observable, and can only be accessed via observation.
- Since the state is not known a priori, the agent has to consider multiple hypotheses about its likely state.
- This is expressed in the **belief state**, which contains a compact representation of the information known to the agent, defined as a probability distribution over possible states.
- Formally, a belief state is a probability distribution over possible states, that is: $b: S \rightarrow [0,1]$
- For a state space of cardinality n, the belief state is therefore represented in a real-valued simplex of dimension (n-1).



Belief state update

- How is the belief state updated over time?
  - Assume that at a given time t, we are in some (hidden) state $s_t = s \in S$. The probability of being in state s at time t is written $b_t(s)$.
  - Based on b_t, the agent selects an action a_t, receives a reward $r(s,a_t)$ and transitions to a new (unobserved) state $s_{t+1} = s'$
  - We then perceive a new observation o_{t+1}, and update the belief state as follows:

$$b_{t+1}(s') = P(s' | o_{t+1}, a_t, b_t)$$

$$\vdots$$

$$= \alpha\, \Omega(o_{t+1}, s', a_t) \sum_{s \in \mathcal{S}} T(s, a_t, s') b_t(s)$$

(where α is a normalisation constant)

Dialogue policies in (PO)MDPs

- Given a POMDP model ⟨S, A, Z, T, Z, R⟩, what action should an agent execute at each time-step?
- We are searching for a function $\pi : B \rightarrow A$, called a policy, which determines the action to perform for each point of the belief space.
  - The policy π thus specifies an action $a = \pi(b)$ for any belief b.
  - defined as a function of a continuous, high-dimensional variable.
- The metric used to evaluate the quality of a policy is called the return. The return is the cumulative, discounted reward:

$$R = \sum_{t=0}^{h} \gamma^t r(s_t, a_t)$$

(where γ is the discount factor and *h* the horizon)

Dialogue policies in (PO)MDPs

- The **value function** V(b) defines the *expected* return of a policy starting at a given position b:

$$V^{\pi}(b) = E\left[\sum_{t=0}^{h} \gamma^{t} r(s_t, a_t) \mid b, \pi\right]$$

- And the optimal policy is simply the policy yielding the highest expected return from the start point b_0:

$$\pi^{*} = \mathrm{argmax}_{\pi}\ V^{\pi}(b_0)$$

- Most reinforcement learning algorithms used to optimise dialogue policies operate by incrementally refining their estimate of the optimal value function V^*

[Sutton, R. S., & Barto, A. G. 1998]

Finding optimal policies

- How to compute good estimates of the optimal value function V*?
- If a full transition model is given: *dynamic programming*
- Else, the agent has to learn from *experience*, by *interacting* with its environment (trial-and-error)
    - Need to balance *exploration* and *exploitation*
    - “Model-based” or “model-free” methods
    - Several algorithms available: Monte Carlo methods, temporal-difference learning, SARSA, etc.
    - Operating with real users or (preferably) with a user simulator

[Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., & Young, S. 2007]

[Henderson, J., Lemon, O., & Georgila, K. 2008]

[Rieser, V., & Lemon, O. 2010]

Outline of the talk

- Background
  - Generalities about dialogue systems
  - Dialogue management, dialogue models
  - Learning dialogue policies
- **Ongoing work**
  - **General vision**
  - Current research directions
  - Implementation & evaluation
- Conclusion

My research objective

- I'm primarily interested in developing spoken dialogue systems for rich, open-ended domains
  - For instance: tutoring systems to foster learning, human-robot interaction for service robots, etc.
  - Go beyond the classical "slot-filling" applications!
- Two issues to solve:
  - High levels of *uncertainty*: speech recognition errors, limited grammar coverage, linguistic or pragmatic ambiguities, etc.
  - Structural *complexity*: dialogue history, task model, external context viewed as *rich relational structures*
- In addition, the dialogue system should be *adaptive* to a variety of internal and external factors

Combining symbolic and statistical methods

- Handcrafted policies have their shortcomings, but so do most learned policies
  - Difficult to integrate "obvious" prior pragmatic knowledge in the system
  - Difficult to manually edit learned policies to modify or extend a given behaviour (without having to relearn everything)
  - Usually operate on crude state representations with very simple user and context models
- Our goal is to find a **hybrid** approach to dialogue management which combines the best of symbolic and statistical methods in a unified framework

[Paek, Tim and Pieraccini, Roberto 2008]

Outline of the talk

- Background
  - Generalities about dialogue systems
  - Dialogue management, dialogue models
  - Learning dialogue policies
- Ongoing work
  - General vision
  - **Current research directions**
  - Implementation & evaluation
- Conclusion

Towards richer state representations

→ Q1: how to formalise the dialogue state?

- One of the first steps of my work will be the formalisation of an adequate representation of the dialogue state
  - Able to capture both rich (ontology-based) relational structures and quantified uncertainty
  - mixture of probabilistic and logical models
- An idea I plan to work on is to use *Probabilistic Description Logics*
  - Description Logics (DL) are ideally suited to describe entities and relations between them... and efficiently reason over these
  - Several probabilistic extensions of DL are available

[Lukasiewicz, T. 2008]

Towards richer state representations

- The introduction of some logic-based representations in the framework would also allow us to perform more sophisticated dialogue state update operations
  - Instead of simple Bayesian (propositional) inference, we could have a *mixture* of Bayesian and Description Logic queries
  - The so-called "user action model" $P(a_u \mid i_u, a_m)$ present in most POMDP-based dialogue managers would particularly benefit from such extension
- Ideally, the dialogue state update operation should be "*anytime*", to be able to react in soft real-time
  - Some sampling-based approximation methods (e.g. particle filters or MCMC) might be necessary

Hybrid dialogue policies

→ Q2: how to formalise hybrid dialogue policies?

- Recent work on "hybrid" dialogue policies has focused on the use of hand-coded policies as a preprocessing filter for the learned policies
- I think we can go much further than that!
- One possible approach would be to use an *interconnected network of concurrent policies* instead of relying a single, monolithic policy.
- The complexity of the interaction would thus be "factored out" by breaking it into smaller, more tractable sub-policies
- These policies would be "connected" with each other, most likely via a global shared state (blackboard architecture)
- Each of these policies could be handcrafted or learned, depending on what is most suited to model the interaction at hand

[Williams, J. D. 2008]

Hybrid dialogue policies

- The policies could also be *hierarchically* connected
  - the output of one policy serving as input to one another
  - Useful to structure the decision-making in several steps
- Another related question is the possibility of introducing *prior pragmatic knowledge* to guide the policy learning process
  - Some possibility about Bayesian model-based reinforcement learning with *model uncertainty*
  - How to specify such knowledge remains unclear

[Cuayáhuitl, H., Renals, S., Lemon, O., & Shimodaira, H. 2010]

[Ross, S. & Pineau, J. 2008]

[Doshi, F., Pineau, J. and Roy, N. 2008]

Hybrid dialogue policies

- Since the state space of our domains is likely to be quite large, the use of function approximators might be necessary
  - The basic intuition behind this is to rewrite the value(-action) function as a parametrized function of the state
  - The function might be anything: linear functions, neural networks, etc.
  - The policy search is then reduced to a search for the optimal parameters (policy-gradient learning)
  - Allow us to generalise / abstract over large state spaces
  - Connection with supervised learning methods
- Idea: try to use *Markov Logic Networks* for function approximation?

[Sutton, R. S., & Barto, A. G. 1998]

[Richardson, M., & Domingos, P. 2006]

Joint optimisations of dialogue processing

→ Q3: how to connect the DM to the rest of the system?

- Typical dialogue systems are structured as a unidirectional pipeline:

ASR → NLU → DM → GEN → TTS

- This is not optimal!
  - Absence of feedback between processes
  - No incremental refinement of interpretations
  - Components are designed in isolation from each other
  - Absence of active runtime "control" over the processes

[Kruijff, G. J., Lison, P., Benjamin, T., Jacobsson, H., & Hawes, N. 2007]

[Brick, T., & Scheutz, M. 2007]

Joint optimisations of dialogue processing

- The ideal solution would be to perform a *joint optimisation* of dialogue understanding, management and generation
  - In order to yield a dialogue system which is globally optimal in respect to a given objective function
  - The problem is of course the "curse of dimensionality" of such optimisation
- An example of how this might work for parsing:
  - parametrise some aspect of the parser and allow the dialogue manager to control their values at runtime in order to dynamically adapt the parser performance.

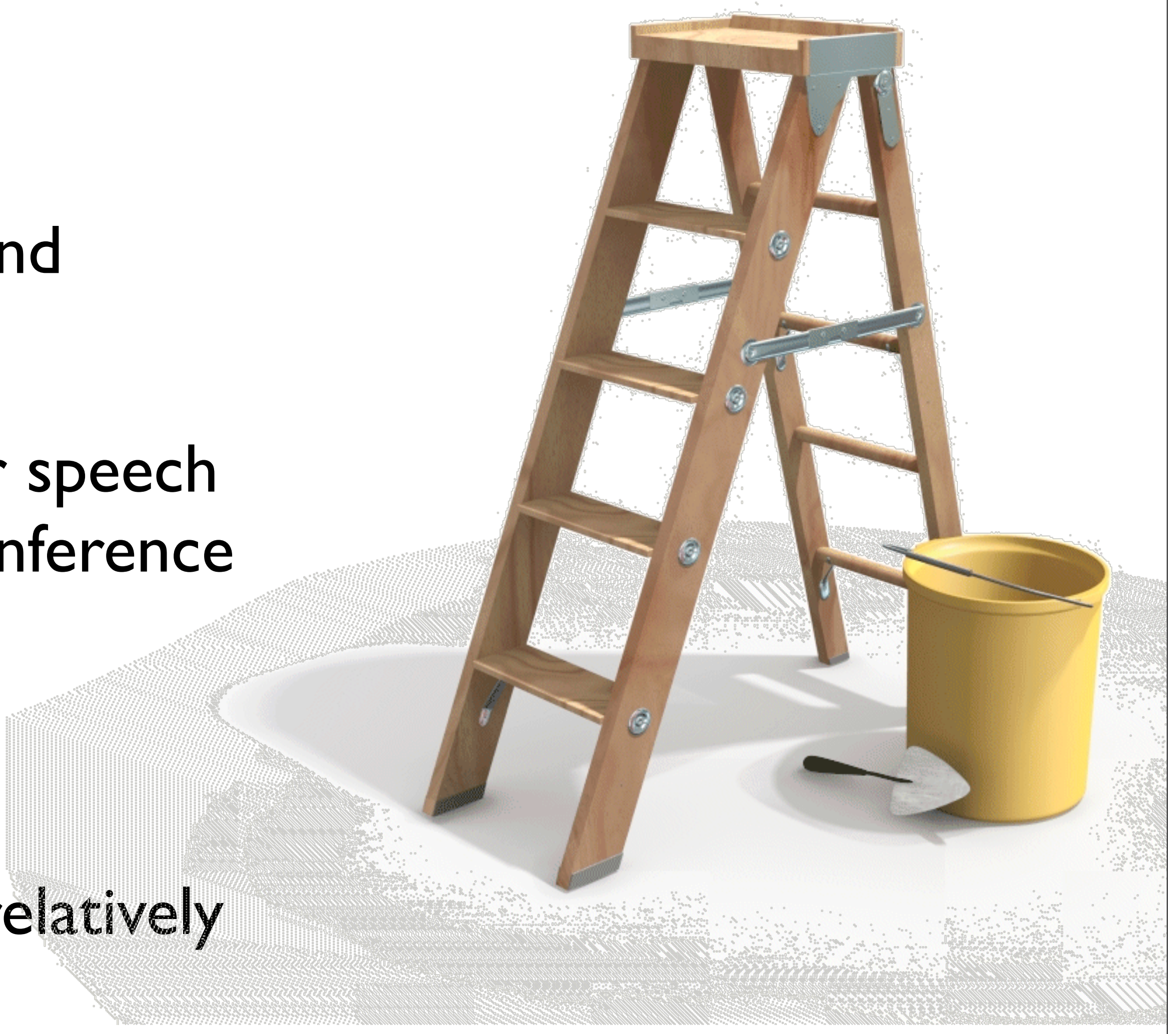
[Lemon, O. 2011 (to appear)]

Outline of the talk

- Background
  - Generalities about dialogue systems
  - Dialogue management, dialogue models
  - Learning dialogue policies
- Ongoing work
  - General vision
  - Current research directions
  - **Implementation & evaluation**
- Conclusion

Implementation plans

- The sketched framework will be implemented as an end-to-end dialogue system for generic domains
  - released under an open source license.
  - Used as a "sandbox" environment to test and evaluate various ideas and algorithms
  - Will include several third-party libraries for speech recognition (Sphinx), synthesis (Mary), DL inference (Pellet), etc.
- *Agile* software methodologies:
  - Iterative & incremental development with relatively short release cycles
  - Test-driven development with unit testing for each component



[PELLET: http://clarkparsia.com/pellet/]

[SPHINX: http://cmusphinx.sourceforge.net/]

[MARY TTS: http://mary.dfki.de]

Empirical evaluation

→ Q4: What would be interesting domains to demonstrate?

- Which application scenario should we choose to demonstrate our approach and gather experimental results?
  - No real data sets for the kind of domains we are interested in
  - Right level of complexity (not too hard, but not too trivial either)
  - Possible ideas: simple tutoring systems, dialogue-based board games, simple human-robot interactions
- Once selected, the application scenarios will be used
  - to gather Wizard-of-Oz data for the policy optimisation
  - and later on to evaluate the system with real-users

[Walker, M.A., Litman, D. J., Kamm, C.A., & Abella, A. 1997 -- PARADISE evaluation framework]

Outline of the talk

- Background
  - Generalities about dialogue systems
  - Dialogue management, dialogue models
  - Learning dialogue policies
- Ongoing work
  - General vision
  - Current research directions
  - Implementation & evaluation
- **Conclusion**

By way of conclusion

- Dialogue management is a complex task!
  - Most work so far has concentrated on slot-filling applications
  - But rich, open-ended domains remain very difficult to model
- I am trying to develop a new hybrid approach to address this issue
  - based on rich representations of context;
  - combining both designed and learned policies in a common framework;
  - and with a tight coupling between dialogue understanding, management and generation
- The approach will be fully implemented and evaluated experimentally in several application scenarios