Building Spoken Dialogue Systems for Embodied Agents
Lecture 2

Johan Bos

School of Informatics

The University of Edinburgh

Outline of the course

- Part I: Natural Language Processing
  - Practical: designing a grammar for a fragment of English in a robot domain
- **Part II: Inference and Interpretation**
  - **Practical: extending the Curt system**
- Part III: Dialogue and Engagement

Yesterday's Lecture

- Given our grammar, how can we systematically assign semantic representations to expressions of natural language

This Lecture

- How can we map natural language utterance to robot primitives
- How can we extract primitives from semantic representations in a systematic way

A: Foundations

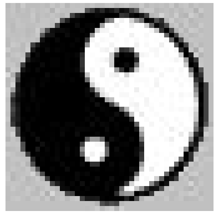
- Let's be serious about semantic interpretation and answer the following questions:
  - Which **semantic formalism** are you going to use?
  - Which **tools for interpretation** are out there?
  - How are you going to **construct representations** for expressions of natural language and deal with ambiguities?

- We will first look at the "tools"

Tools for Semantic Interpretation

- **Theorem Proving**
  – Useful for drawing inferences, such as checking for inconsistencies or informativeness
- **Model Building** (Model Generation)
  – Useful for checking consistency and building a discourse model
- **Model Checking**
  – Useful for querying properties of the constructed discourse model

The Yin and Yang of Inference



- **Theorem Proving** and **Model Building** function as *opposite forces*
- Suppose φ, a logical formula, representing a certain discourse δ
  - If a theorem prover succeeds in finding a **proof** for ¬φ, then δ is **inconsistent**
  - If a model builder succeeds to construct a **model** for φ, then δ is **consistent**

Using Model Building

- Example: “I want to fly from Stansted to Paris”
- Formula (First-order logic):
  - $\exists e(fly(e)\&agent(e,i)\&from(e,stansted)\&to(e,paris))$
- Axioms (travel domain)
  - $\forall x\forall e\forall z(fly(e)\&agent(e,x)\&to(e,z)\rightarrow destination(x,z))$
  - $\forall x\forall e\forall z(fly(e)\&agent(e,x)\&from(e,z)\rightarrow origin(x,z))$
  - And so on…
- Model (D the domain, F the interpretation function):
  - $D=\{d1,d2,d3\}$
  - $F(i)=d1$, $F(stansted)=d2$, $F(paris)=d3$, $F(destination)=\{d1,d2)\}$, $F(origin)=\{d1,d3\},\cdots$.

Model Checking

- A Model Checker (for FOL) is a tool that checks whether a certain model satisfies certain propositions
- Almost like asking a yes-no question
  - Example: are ‘walkers’ the same as persons?
  - Query: satisfy(∀x(walk(x)↔person(x)),M,[]).
- Can also be used to extract information
- Similar to asking a wh-question
  - Example: who is a ‘walker’?
  - Query: satisfy(walk(x),M,[g(x,Answer)]).

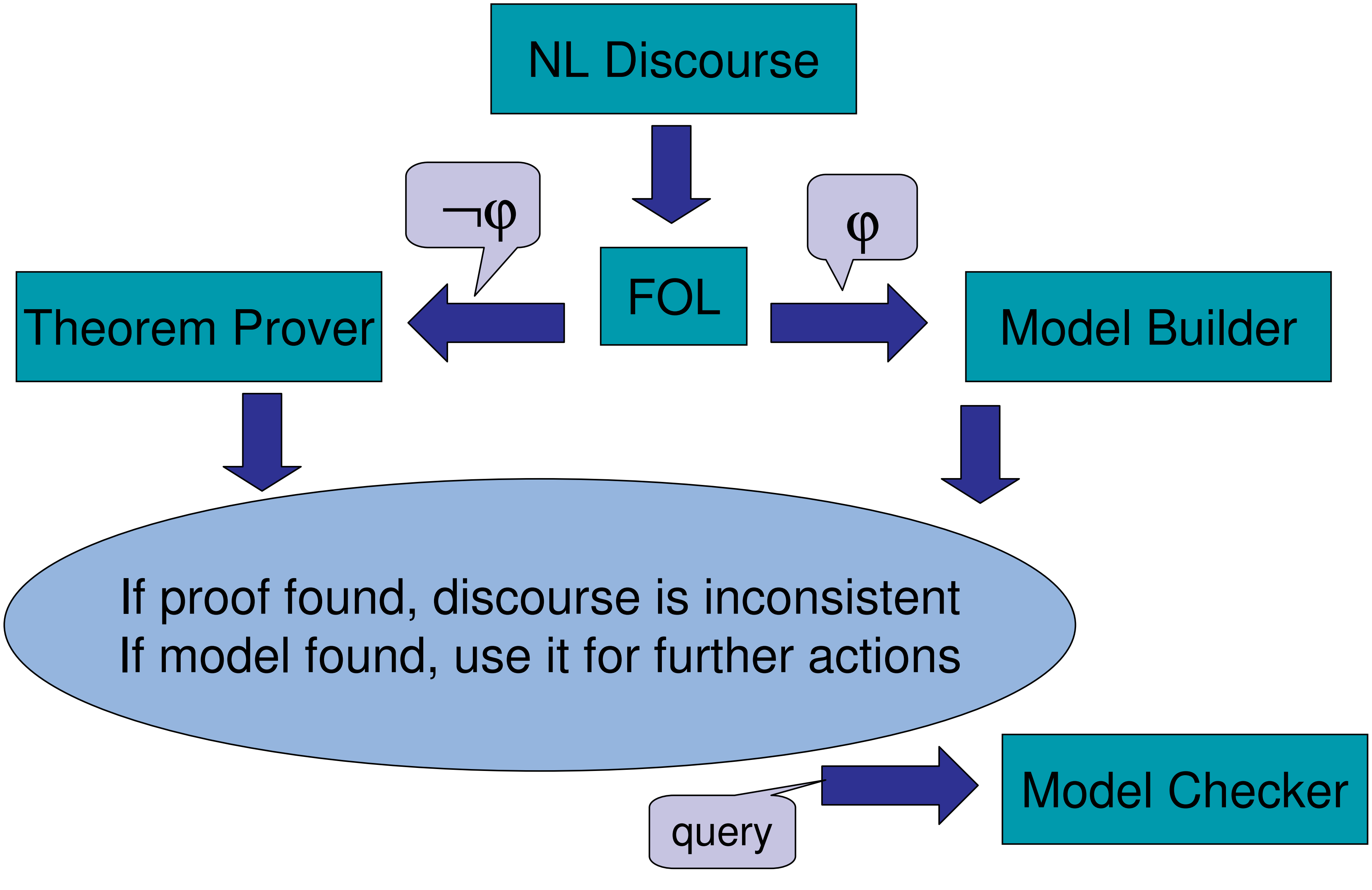
The Beauty of Finite Models

- Minimal (no redundant information)
- Flat (no recursion)
- Deals naturally with quantification, disjunction, conditionals, negation
  - "I want to fly from either Stansted or Luton, but not from Stansted on Fridays"
- Model Checking tools available
- Useful for many NLP tasks:
  - Question answering
  - Disambiguation
  - Interpretation of Instructions

Now we know what tools are available, what is a sensible choice for semantic formalism?

- First-order logic
  - A lot of tools out there, but relatively bad computational properties
- Higher-order logic
  - Very expressive, but currently no useful inference tools
- Description logics
  - Relatively good computational properties, but limited expressive power

The overall picture:



B: Applications

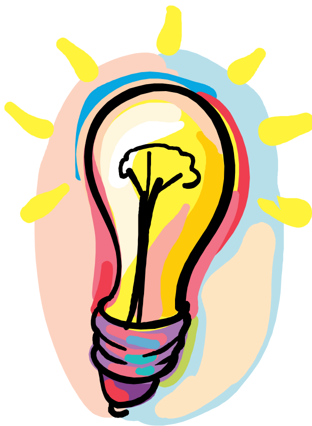
- Spoken Dialogue Systems with Embodied Agents (Small domains, hence feasible)
- Example Applications:
  - Home Automation
  - Godot the Robot
  - Curt (robot simulation)
- Three Examples of the Beauty of Models
- Conclusions

Example Application: Home Automation

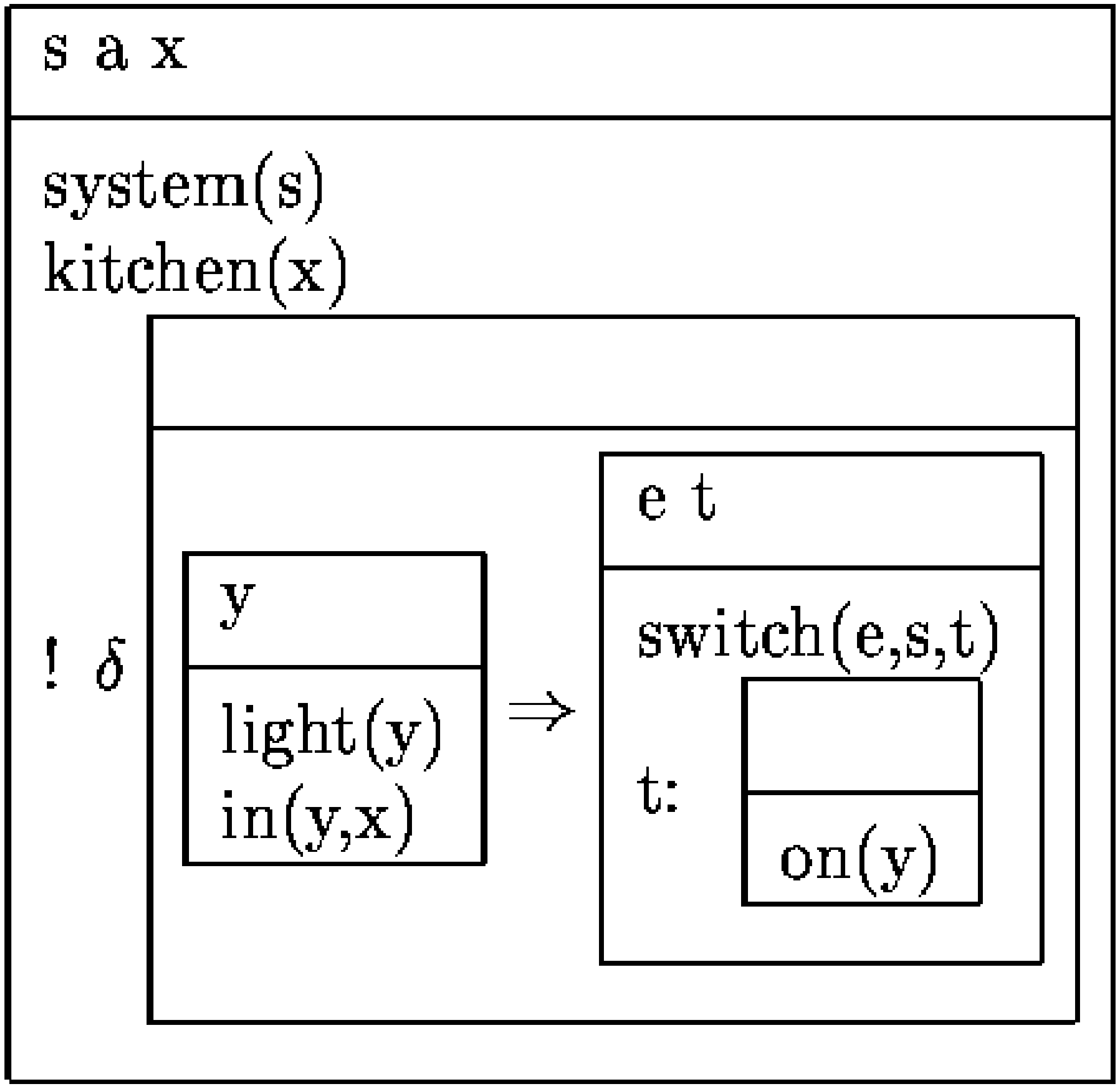
- Implemented as society of OAA agents:
  - ASR (speech recognition): NUANCE
  - SYN (synthesis): FESTIVAL
  - RES (resolution): DORIS
  - INF (inference): SPASS, MACE
- XML configuration of domain knowledge
- Application: Home Automation
  - X-10 and HEYU
  - Lights and Radio in ‘Smart Office’

Beauty of Models
Example 1: Quantification

- Example Instructions:
  - "Clean all the rooms on the first floor!" (Robot)
  - "Turn of every light except the light in the kitchen!" (Home Automation)
- Model builder will produce a model with the number of primitives satisfied by the domain of quantification



Example DRS



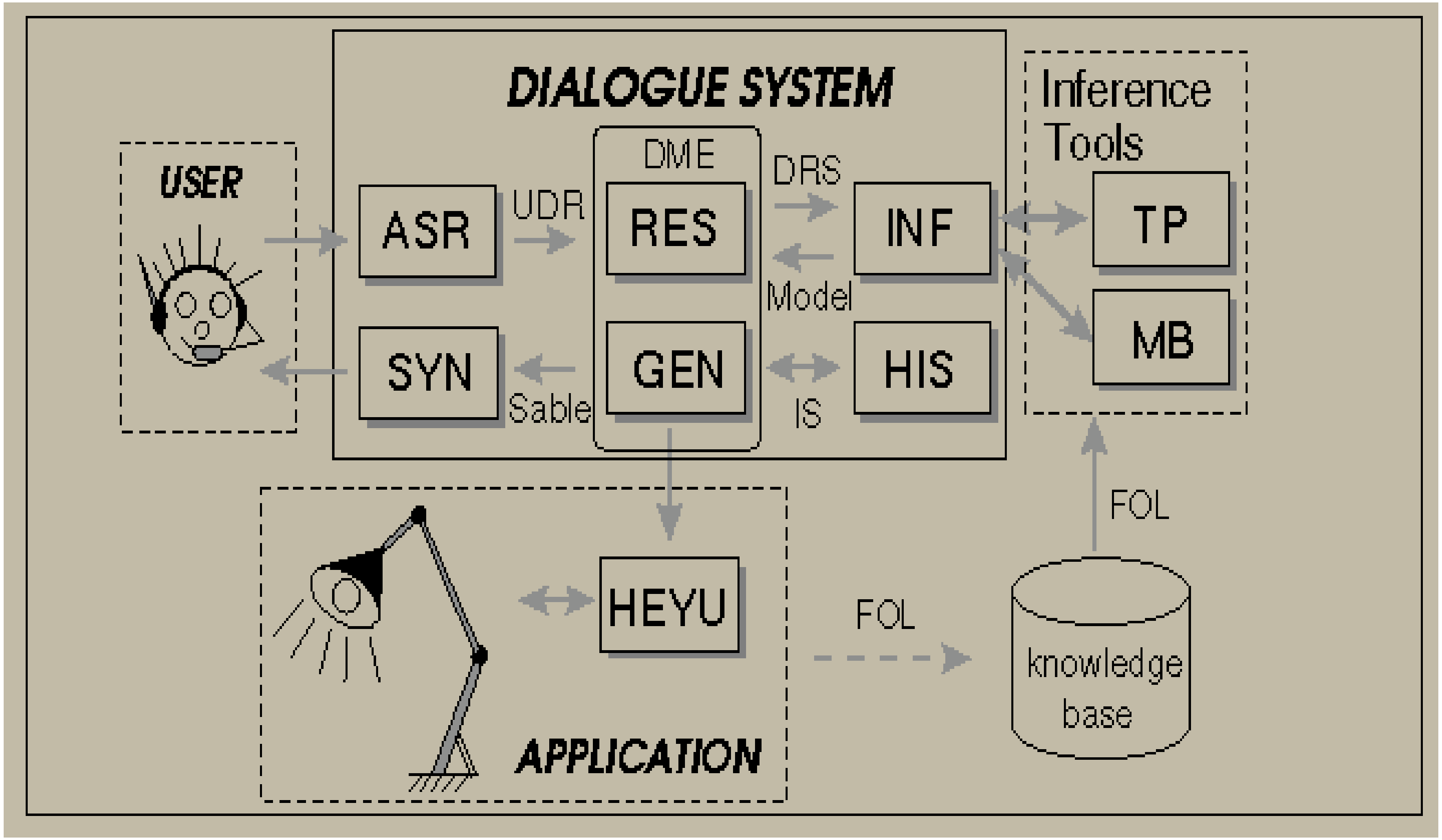
- DRS for 'Switch every light in the kitchen on'

Example: First-Order Model

- Instruction: “Switch every light in the kitchen on!”
  - $\exists w\ \exists s\ \exists x(\text{possible-world}(x)\ \&\ \text{system}(w,s)\ \&\ \text{kitchen}(w,x)\ \&\ \exists v \exists a\ (\text{action}(w,a,v)\ \&\ \forall y\ (\text{light}(a,y)\ \&\ \text{in}(a,y,x) \rightarrow \exists e \exists t(\text{switch}(w,e,s,t)\ \&\ \text{on}(t,y))))))$
  - Output model:

```
D={d1,d2,d3,d4,d5,d6,d7,d8}
F(possible_world)={d1,d2,d3}
F(system)={(d1,d4),(d2,d4),(d3,d4)}
F(kitchen)={(d1,d5),(d2,d5),(d3,d5)}
F(action)={(d1,d2,d3)}
F(light)={(d1,d6),(d2,d6),(d3,d6),(d1,d7),(d2,d7),(d3,d7)}
F(in)={(d1,d6,d5),(d2,d6,d5),(d3,d6,d5),(d1,d7,d5),(d2,d7,d5),(d3,d7,d5)}
F(poweron={(d2,d6),(d2,d7)}
F(off)={(d1,d6),(d1,d7)}
F(on)={(d3,d6),(d3,d7)}
```

Architecture



Beauty of Models
Example 2: Question Answering

- Translate a question ?x(Rx,Sx) into:
  - ∀x((Rx&Sx)↔pos-answer(x))
  - ∀x((Rx&¬Sx) ↔neg-answer(x))
- Example: where are you going to?
  - ?x(location(x),∃e(go(e)&agent(e,u)&to(e,x)))
- Give this to a model builder
  - The denotations of pos-answer and neg-answer provide the answer to the question
- Use a model checker to generate answer

Beauty of Models

Example 3: Learning new Information

- Information of the environment stored in the robot's internal database
- Example Instructions:
  - "Go to the corridor!"
  - "You are in the corridor leading to the emergency exit."
- Model provides more information than the robot's internal database -- update

Summary

- Tools for computational semantics
  - theorem proving
  - model building
  - model checking
- Use first-order logic
- Role of model building

Conclusion:
Inference in Dialogue Systems

- Slot-filler or frame-based approaches are old-fashioned
- Model building provides an alternative, opening a wide variety of interpretation tasks
- Three reasons for first-order theorem proving to play a role in future systems
  - Theorem proving is still a promising emerging field
  - Not tuned to linguistic problems
  - Current approach is non-incremental

Practical Session

- The CURT system
- See www.comsem.org
  (Blackburn & Bos)
- File: robotCurt.pl