Language Based Communication between Humans and Robots

Alexander I. Rudnicky

MSR Faculty Summit

13 July 2010

Carnegie Mellon

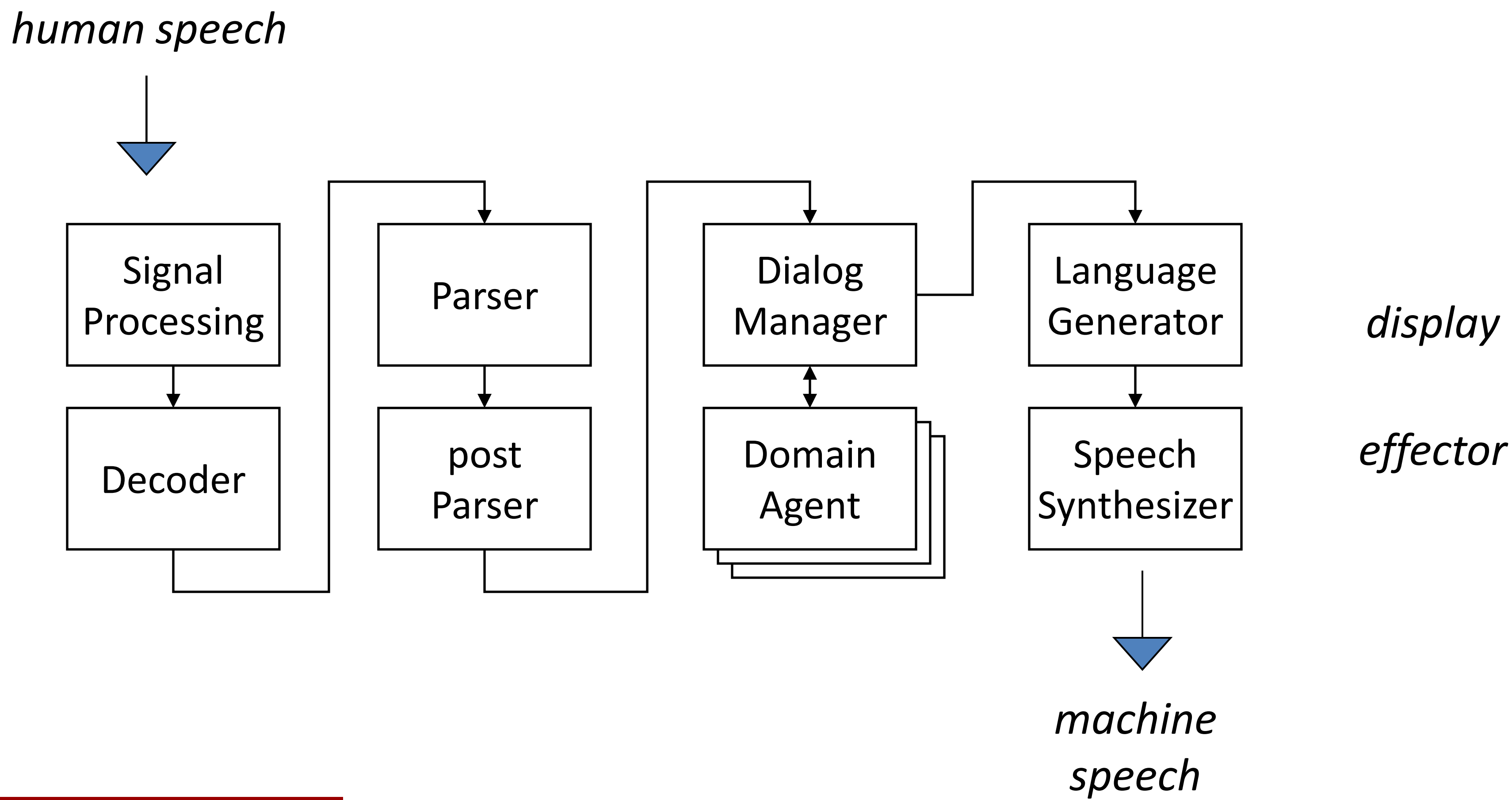
Some areas of application

- Search and rescue
  - Mobile teams of humans and robots
- Maintenance and logistics
  - DARPA’s mule / big dog
- Domestic robots
  - Companions and simple tasks

Spoken Dialog Systems

- Calculator and Calendar (1988)
  - Video link
- Office Manager (1990)
- Communicator (~2000)
- TeamTalk (2005-)

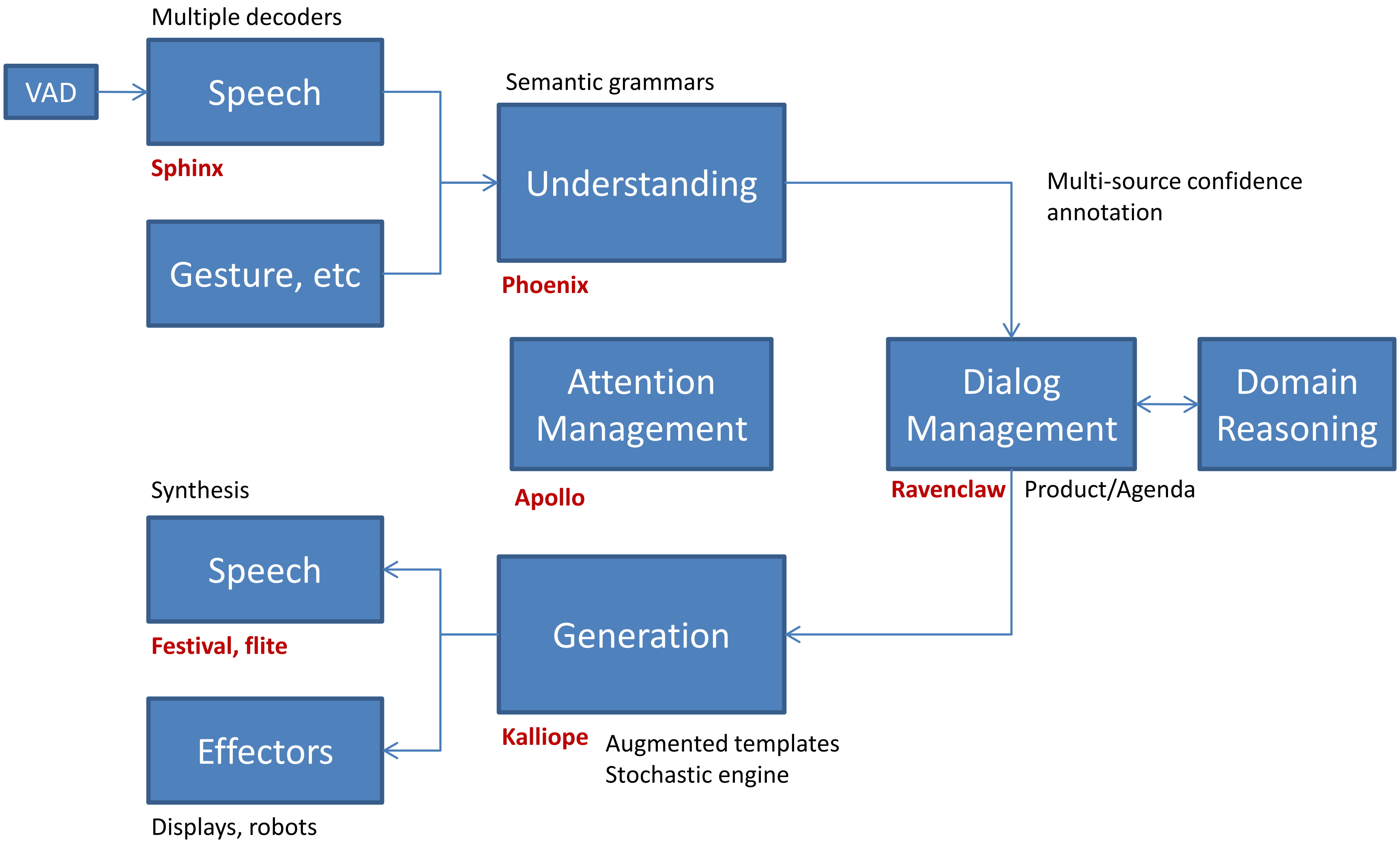
A generic speech system



Core problems in spoken language interaction

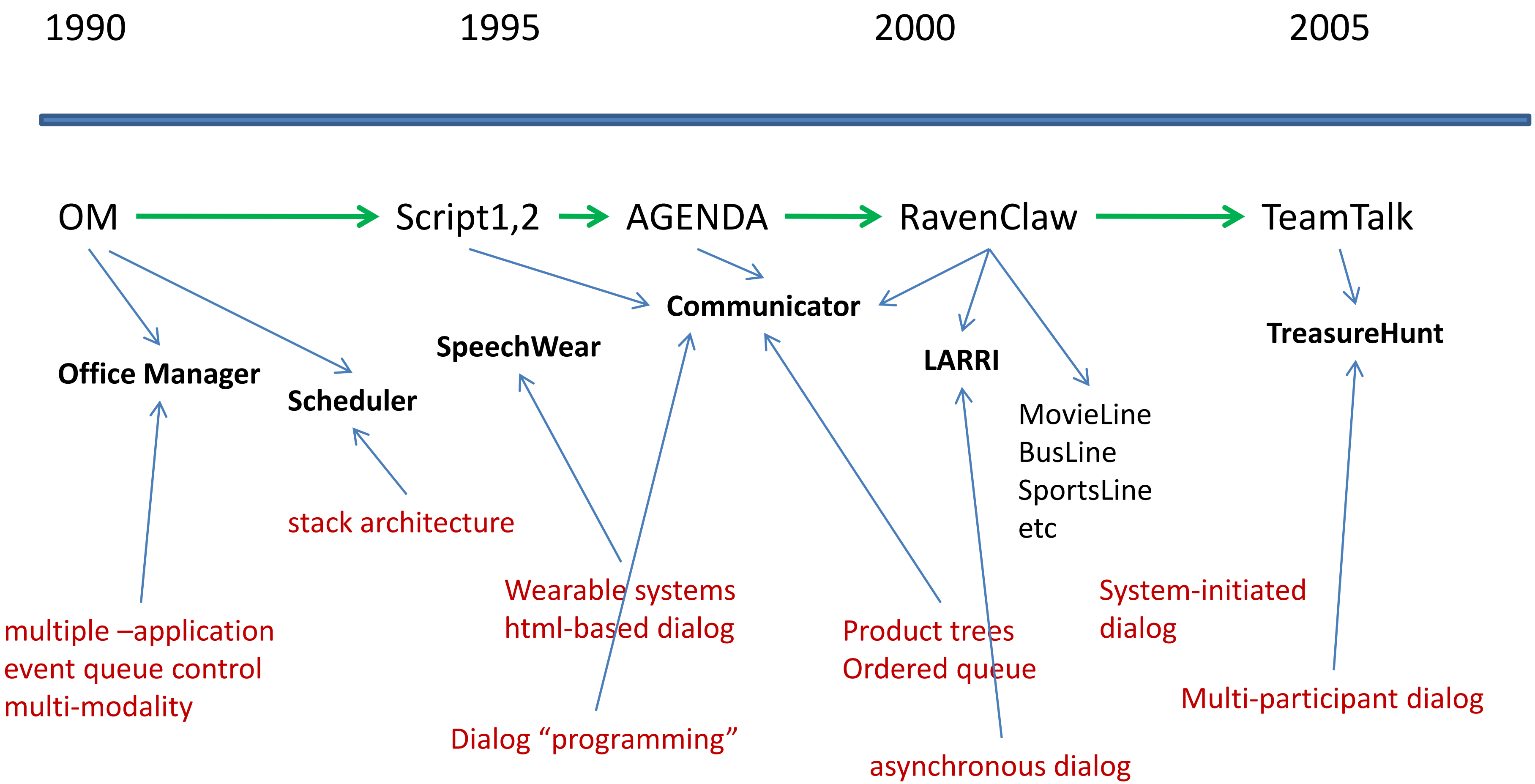
- Recognition
  - ***Mapping speech to words***
  - Tracking the topic and dynamically changing the language
- Understanding
  - ***Words to concepts***
  - Capturing domain-specific natural language
- Dialog Management
  - ***Creating a coherent conversation***
  - Tracking the topic and supporting mixed-initiative dialog
- Generation
  - ***Mapping concepts to words***
  - Creating comprehensible templates; learning from corpora
- Synthesis
  - ***Understandable voices***
  - Limited-domain synthesis
- Domain Reasoning
  - ***Capturing real-world information***
  - Plans, ontology-based reasoning

Dialog system architecture



Carnegie Mellon

Speech applications and dialog systems at Carnegie Mellon



Carnegie Mellon

Spoken Dialog Systems

- Calculator and Calendar (1988)
- Office Manager (1990)
  - Video link
- Communicator (~2000)
- TeamTalk (2005-)

Key ideas

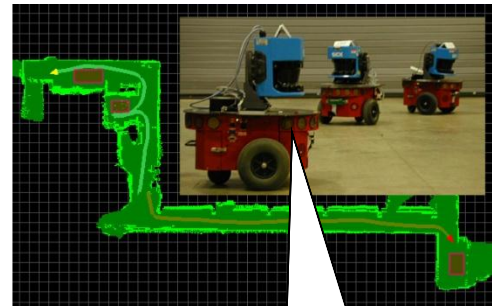
- **Product / Agenda control**
  - Tasks represented as (partial or complete) plans
  - Mixed-initiative dialogue
- **Plans and plan composition**
  - Tasks represented as hierarchical plans
  - Sub-plan library for dynamically modifying plan
- **Separation of domain and discourse processing**
  - Domain Reasoner to maintain context, interact with the world ("back-end")
- **Concept Based dialog flow**
  - Concept-in / concept-out; no explicit understanding at the dialog level
- **Separation of task and domain knowledge**
  - Error-recovery sub-dialogs

Spoken Dialog Systems

- Calculator and Calendar (1988)
- Office Manager (1990)
- Communicator (~2000)
  - Video Link
- TeamTalk (2005-)

Domain: *Treasure Hunt*

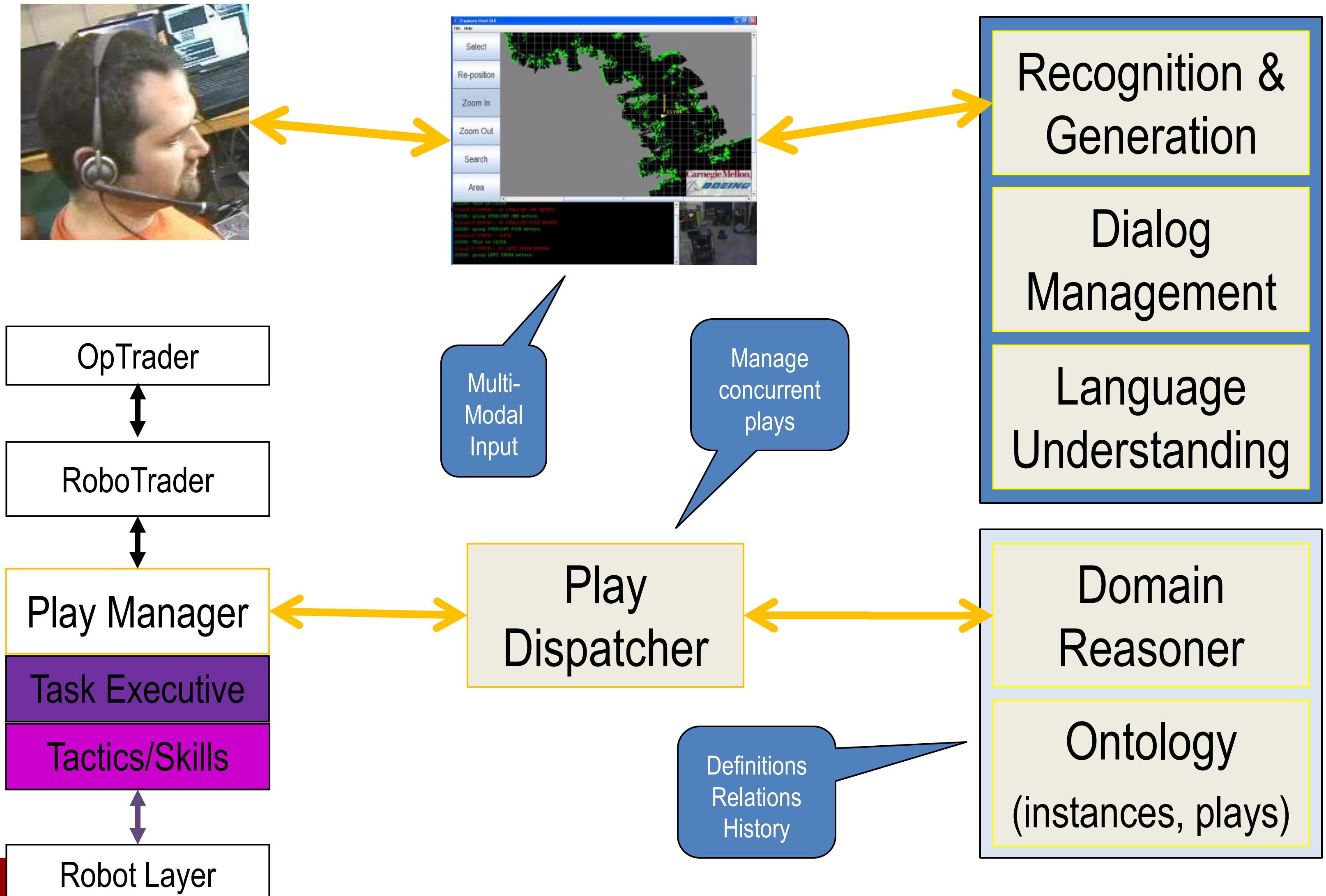
Someone search area 3.
Robot X retrieve treasure
from position Y



Let's form a
sub-team…

Human-robot teams coordinate to explore and locate items in an *unknown* environment

Architecture



Issues in Human-Robot Dialog

- Managing multi-participant dialog
  - Addressing, turn-taking, controlling the floor
  - Communicating robot state, urgency
- Integrating multiple human and robot actors into the same communication space
  - Channel maitenance, interruption, eaves-dropping
- Grounding and sharing ontologies
  - Talking about physical environments, actions
  - Augmenting language and concepts
- Sharing knowledge
  - Mapping human instructions to robot representation and action
  - Learning new action sequences

TeamTalk Research Issues

- **Grounding**
  - How humans and robots can agree about things in the environment
- **Instruction**
  - Allowing robots to learn through instruction by humans
- **Spatial language**
  - Communicating about objects and events in the world
- **Multi-participant dialog management**
  - Using spoken language in groups

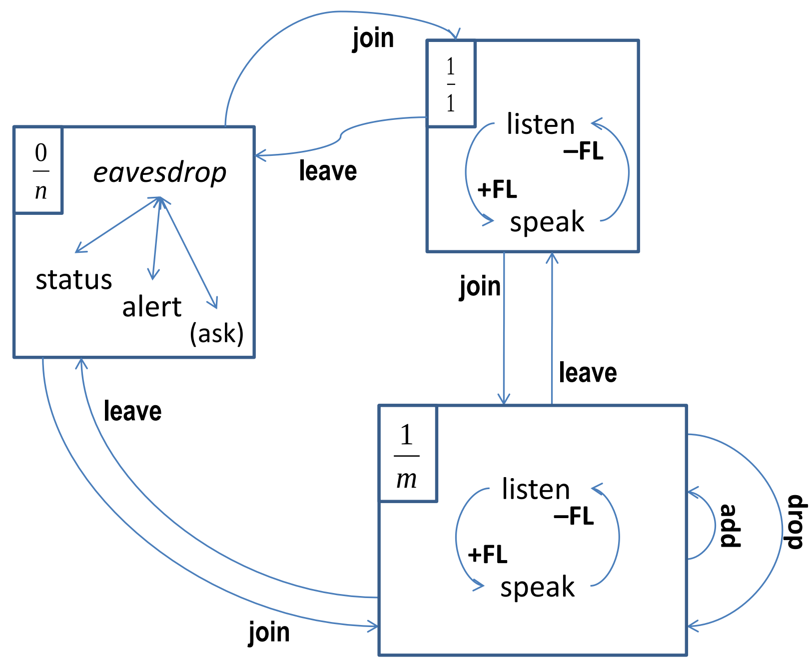
Spoken Dialog Systems

- Calculator and Calendar (1988)
- Office Manager (1990)
- Communicator (~2000)
- TeamTalk (2005-)
  - Video link

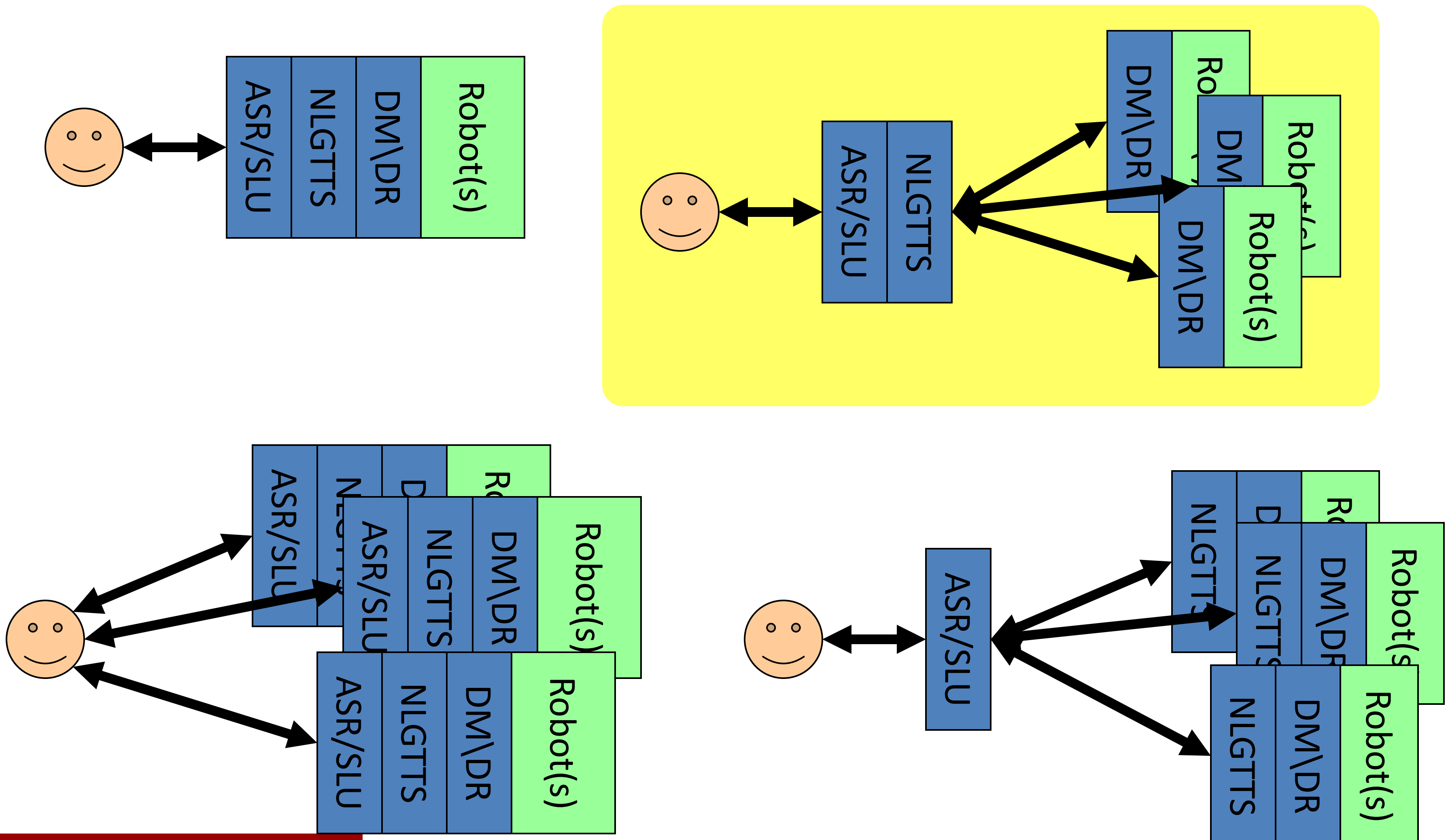
Carnegie Mellon

Managing Dialog

- Managing one's own behavior in conversation
  - Humans know how to do it
  - Robots don't
    - Build a computational model of multi-participant dialog



Some possible dialog system architectures



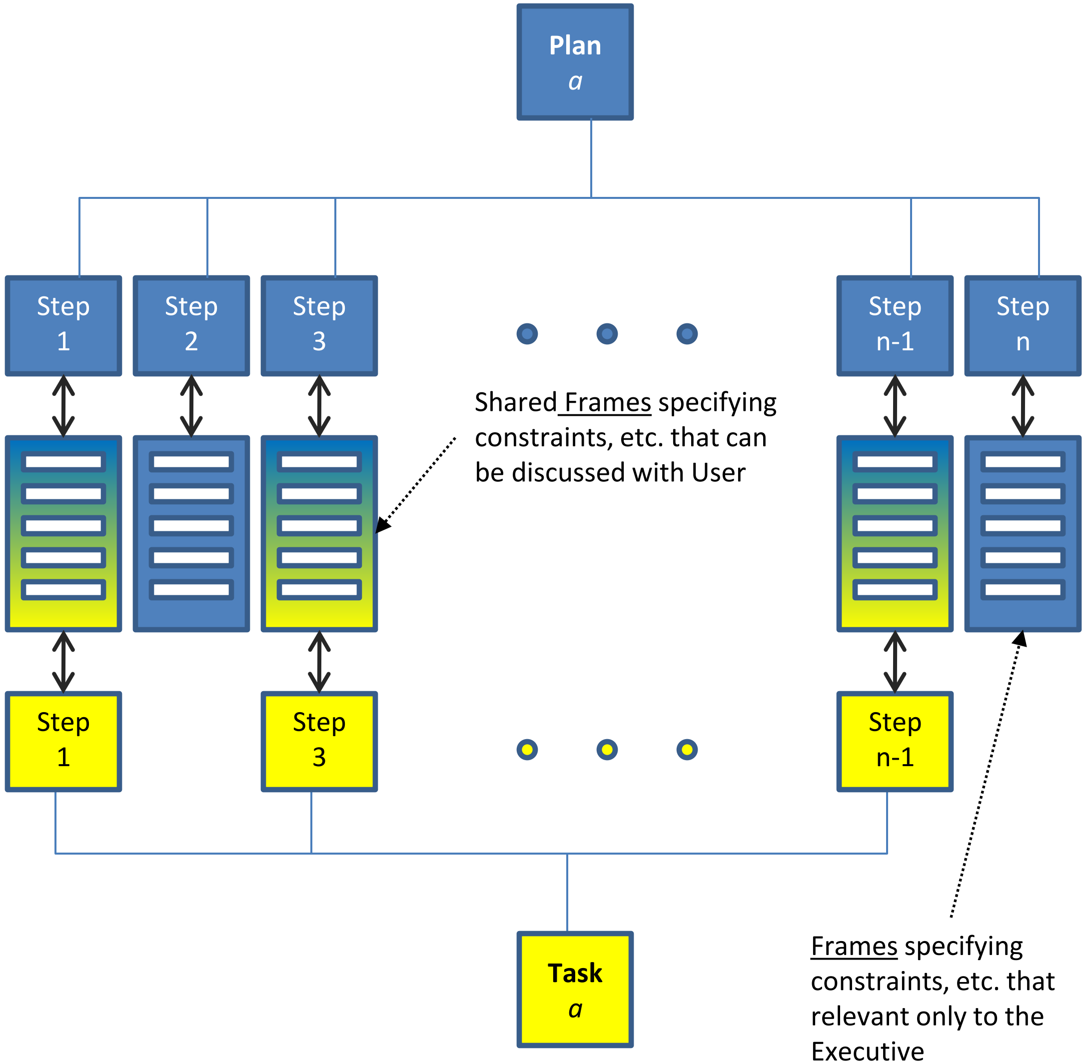
Managing Knowledge

- Robots build and maintain a shared world model
  - OWL-based ontology knowledge base
- Humans introduce and define new entities
  - Locations, plans, etc
  - Robots interactively build and clarify entities
  - Augment ontology as needed

Interaction between Dialog and Robot Components
OLY ↔ MAP ↔ EXE structure

The executive and Olympus each manage corresponding plans. The correspondence is managed by the Mapper. The Executive's plans are complete with respect to the task at hand. Olympus only knows about those parts of the plan that humans may be expected to be able to contribute to.
The frames in the diagram specify the <u>slots</u> needed to carry out the plan. Not all frames, or slots within a frame need to be exposed to Olympus.

Olympus maintains a live list of all possible tasks. Whenever the User selects a task (through a request or command) Olympus interacts with the user to fill in all <u>required</u> slots. Once this is completed the plan is shared with the Executive, which then attempts the plan. If problems come up which require user intervention the plan is suitably <u>annotated</u> and Olympus notified (through the Mapper). After interacting with the User Olympus returns what it believes is a consistent set of updated slots. The Executive then attempts to execute the new plan. The user may be consulted again, as problems develop.



Carnegie Mellon

Managing Learning

- Articulating the learning cycle
  - Sketch → Detail → Exceptions
- Interactive learning
  - Clarification
  - Modification
- Maintaining knowledge over time
  - Consistency
  - Generalization
  - Forgetting

Conclusion

- Moving spoken language systems into the world
  - Uncertainty about the state of the world and other participants
  - Learning through language and interaction
  - Expanding the architecture