

# The Agent-based Dialogue Management Platform

Mark Buckley

`markb@ags.uni-sb.de`

Theorema-Ultra-Omega '05 Workshop

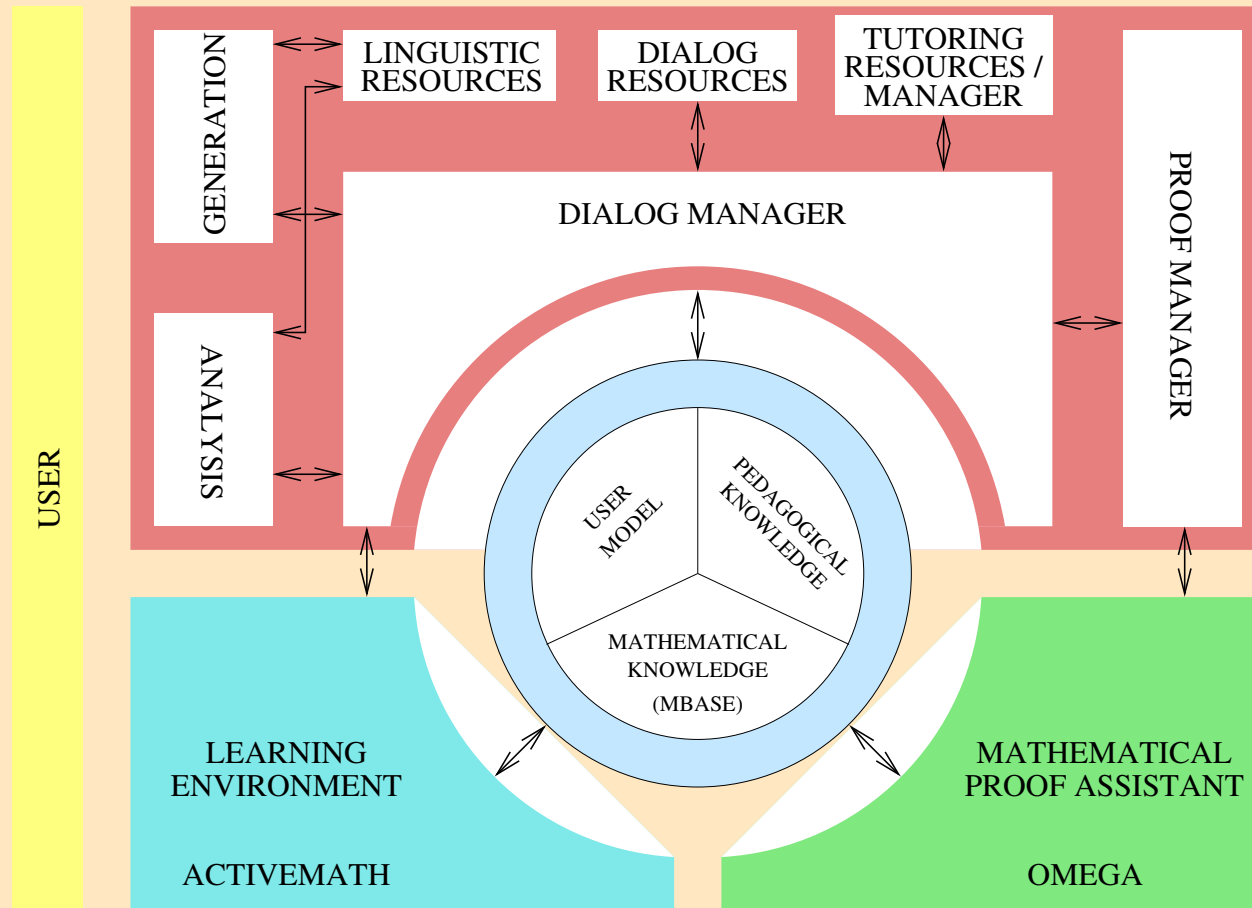
15 November, 2005

# The DIALOG Project

► Dialogue Management

►  $\Omega$ -Ants

► Agent-based Dialogue Manager



# Role of the Dialogue Manager



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

Function:

- Controls **dialogue flow**
- **Integrates** system modules
- Maintains **dialogue context**

# Role of the Dialogue Manager



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

Function:

- Controls **dialogue flow**
- **Integrates** system modules
- Maintains **dialogue context**
  - ▶ Description of state of the dialogue
  - ▶ Events change the state:
    - user: *I don't understand your question.*

# Information-State Update Approach



---

▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

---

Framework for theories of dialogue management

# Information-State Update Approach



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

Framework for theories of dialogue management

- Information State

- ▶ Representation of dialogue context
- ▶ Storage of shared data

# Information-State Update Approach



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

Framework for theories of dialogue management

## ■ Information State

- ▶ Representation of dialogue context
- ▶ Storage of shared data

## ■ Dialogue moves

- ▶ Representation of the function of utterances

# Information-State Update Approach



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

Framework for theories of dialogue management

## ■ Information State

- ▶ Representation of dialogue context
- ▶ Storage of shared data

## ■ Dialogue moves

- ▶ Representation of the function of utterances

## ■ Update rules

- ▶ Transitions between Information States

# Information-State Update Approach



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

Framework for theories of dialogue management

## ■ Information State

- ▶ Representation of dialogue context
- ▶ Storage of shared data

## ■ Dialogue moves

- ▶ Representation of the function of utterances

## ■ Update rules

- ▶ Transitions between Information States

## ■ Update strategy

- ▶ Chooses update rules

# Design of the Dialogue Manager?



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

Needs to provide:

- Integration of **system modules**
- A central **information state** with direct **access**
- Flexible **control**

# Design of the Dialogue Manager?



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

Needs to provide:

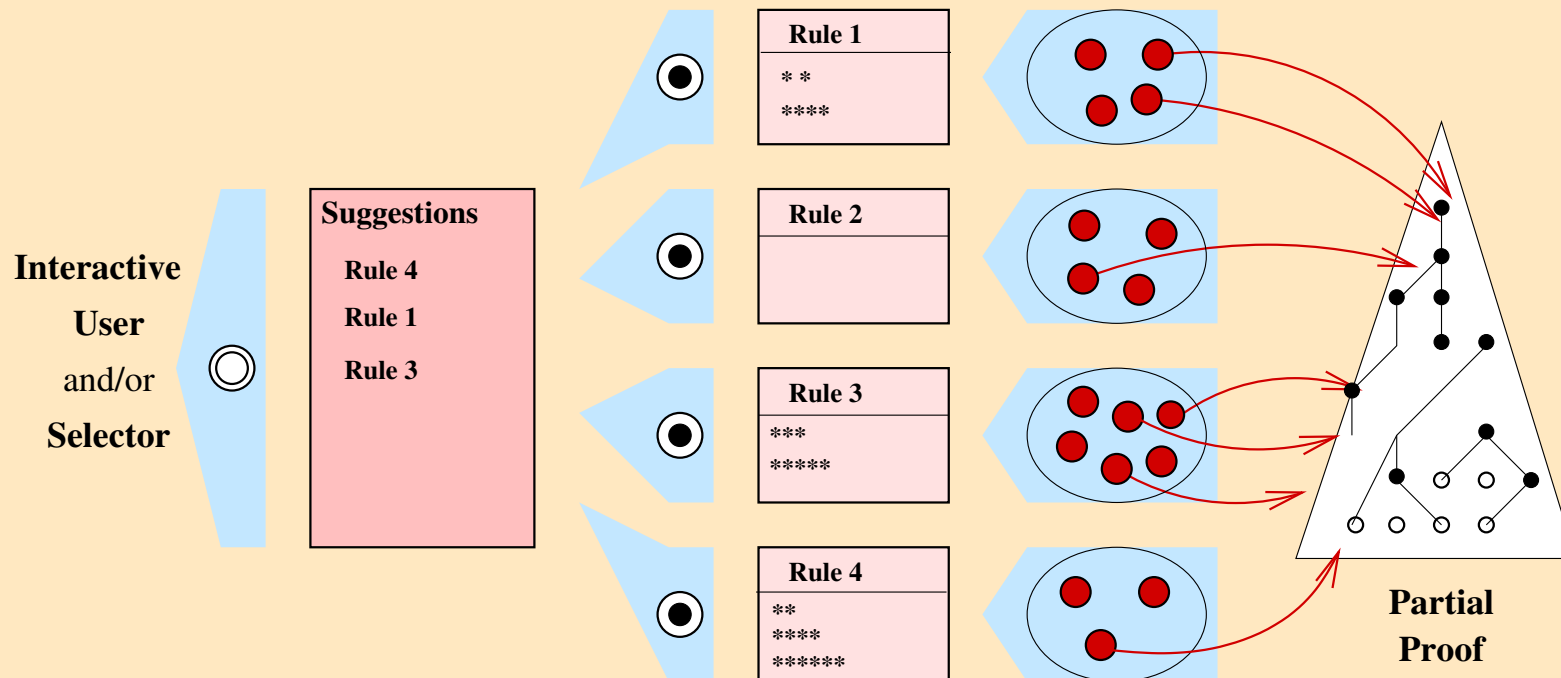
- Integration of **system modules**
- A central **information state** with direct **access**
- Flexible **control**

**Solution:** Borrow from the design of  $\Omega$ -Ants!

- Represent update rules and modules by **software agents**

A **suggestion mechanism** for interactive proof planning in  $\Omega_{MEGA}$

- **Concurrent agents** represent  $\Omega_{MEGA}$  proof planning commands



$\Omega$ -Ants binds in **external systems** like computer algebra systems

- The command

$$\frac{Premlist}{Conc} Otter$$

is represented by an  $\Omega$ -Ants agent

Ω-Ants binds in **external systems** like computer algebra systems

- The command

$$\frac{Premlist}{Conc} Otter$$

is represented by an Ω-Ants agent

**Benefits** of Ω-Ants:

- Concurrent agent execution
- Runtime flexibility
- Resource adaptiveness

# An Agent-based Dialogue Manager



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

- A **platform** for dialogue management
  - ▶ Build dialogue manager on top of this
- Use agent-based techniques from  $\Omega$ -Ants

# An Agent-based Dialogue Manager



► Dialogue Management

►  $\Omega$ -Ants

► Agent-based Dialogue Manager

- A **platform** for dialogue management
  - Build dialogue manager on top of this
- Use agent-based techniques from  $\Omega$ -Ants

It provides:

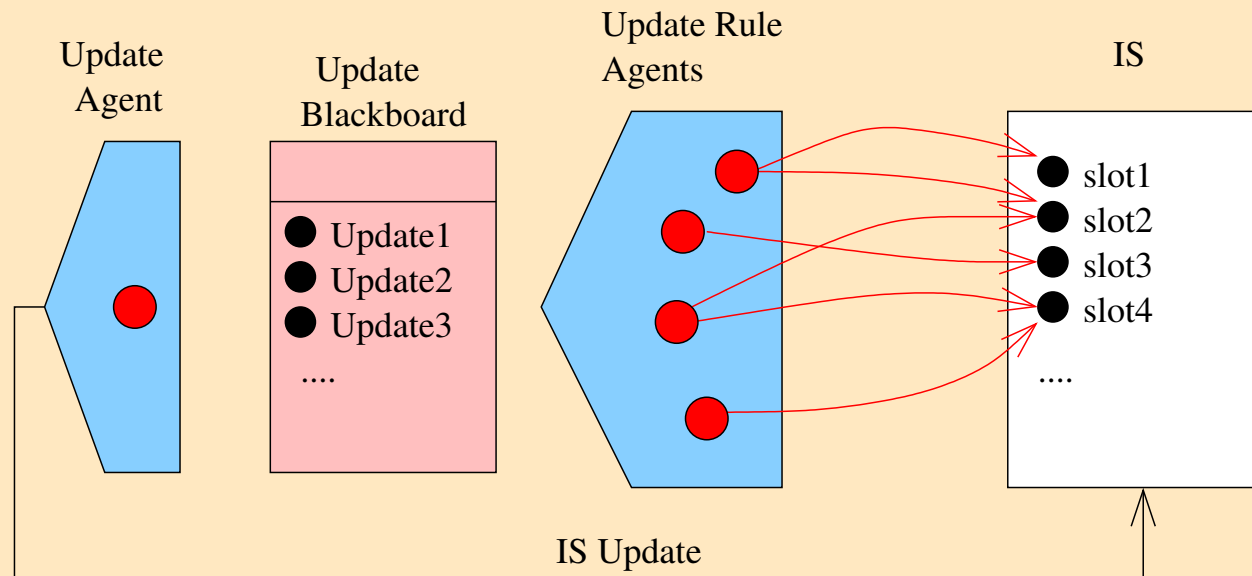
- An **information state** (like  $\Omega$ -Ants' *PDS*)
- Integration of **modules** (like  $\Omega$ -Ants' external systems)
- **Concurrent execution** by software agents
- A **meta-level** due to hierarchical design
- **Reusability** due to domain independence

# Architecture

► Dialogue Management

►  $\Omega$ -Ants

► Agent-based Dialogue Manager



# Defining a Dialogue Manager



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

## ■ Information State

- ▶ A set of **named, typed slots**
- ▶ Readable by update rules and writable by the update agent

# Defining a Dialogue Manager



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

## ■ Information State

- ▶ A set of **named, typed slots**
- ▶ Readable by update rules and writable by the update agent

## ■ Update Rules

- ▶ Compute **updates** of IS slots
- ▶ Consist of preconditions, sideconditions, effects:

$$\frac{\{(s_1, b_1), \dots, (s_j, b_j)\}}{\{(s_1, f_1), \dots, (s_l, f_l)\}}_n < (v_1, f_1), \dots, (v_k, f_k) >$$

# Update Rule Execution



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

- Current IS **satisfies** the preconditions  $\Rightarrow$  Rule can fire
  - ▶ Sidecondition expressions are evaluated and bound to variables
  - ▶ Effect expressions evaluated
  - ▶ **IS update** is computed and written to update blackboard

# Update Rule Execution



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

- Current IS **satisfies** the preconditions  $\Rightarrow$  Rule can fire
  - ▶ Sidecondition expressions are evaluated and bound to variables
  - ▶ Effect expressions evaluated
  - ▶ **IS update** is computed and written to update blackboard

Example:

$$\frac{\{stringp(utterance)\}}{\{(linguistic\_meaning := result)\}} NLAnalyser < (result := call\_to\_NL\_module(utterance)) >$$

# Update Rule Execution



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

- Current IS **satisfies** the preconditions  $\Rightarrow$  Rule can fire
  - ▶ Sidecondition expressions are evaluated and bound to variables
  - ▶ Effect expressions evaluated
  - ▶ **IS update** is computed and written to update blackboard

Example:

$$\frac{\{stringp(utterance)\}}{\{(linguistic\_meaning := result)\}} NLAnalyser < (result := call\_to\_NL\_module(utterance)) >$$

- Update agent monitors update blackboard & executes update
- Other agents see change in IS
- A full transition has been made

# Benefits of new Dialogue Manager



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

What do we gain from this approach?

- Benefits from  $\Omega$ -Ants: Concurrency, flexibility
- A natural way to integrate external systems
- Application of heuristics in update strategy
- Reasoning on instantiated updates, not rule definitions
  - ▶ Dialogue Manager can control top-level system execution
- Better support for interleaving system modules

# Conclusion



▶ Dialogue Management

▶  $\Omega$ -Ants

▶ Agent-based Dialogue Manager

- An ISU dialogue management platform using  $\Omega$ -Ants philosophy
- Facilitates other research directions in DIALOG
  - ▶ Interleaving of NLU, domain reasoning, tutorial reasoning
- General Architecture: Reusable and adaptable