



Universidad  
de Alcalá

# Autonomous Control Architectures

---



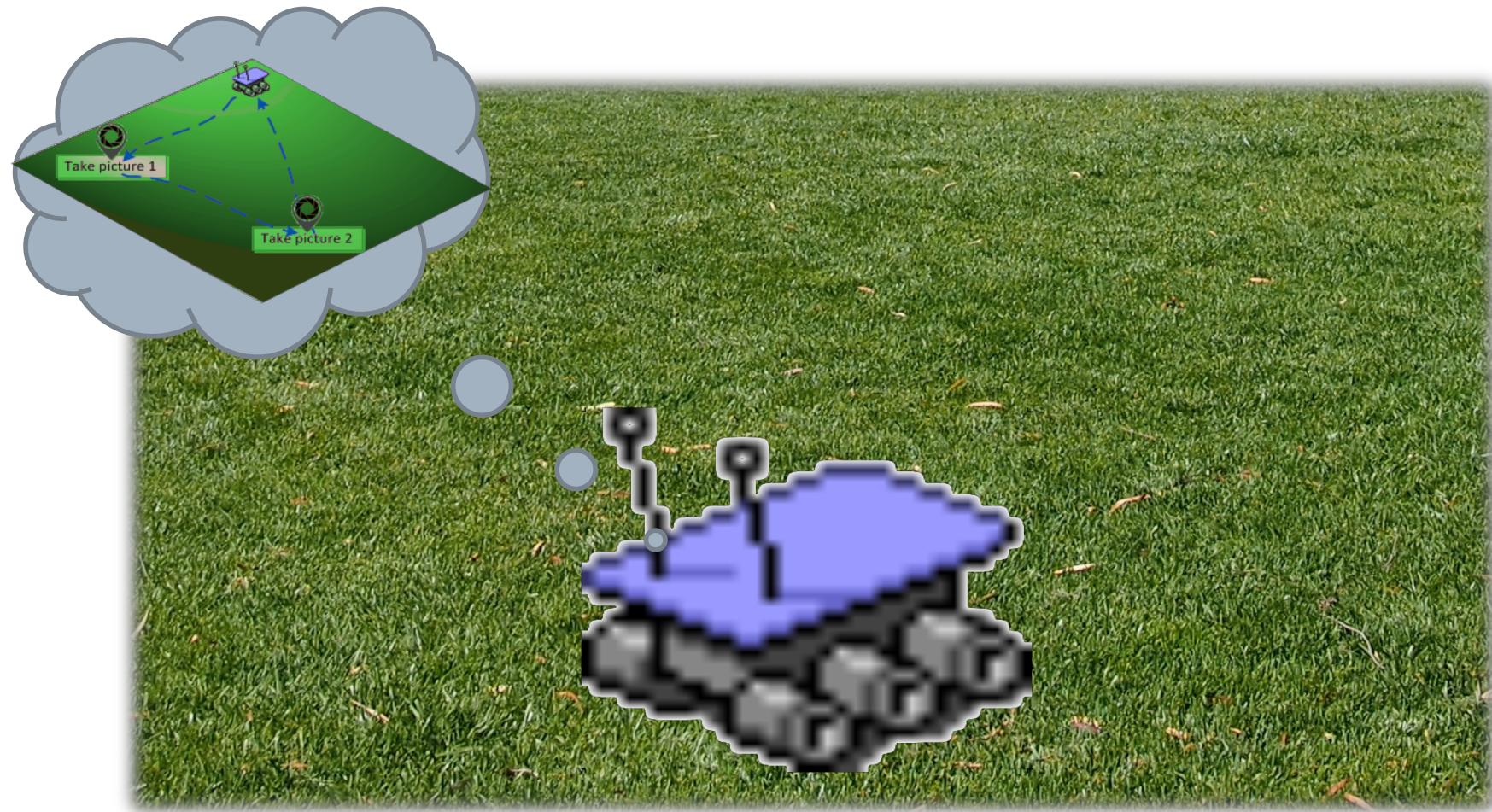
# Outline

---

- **Introduction**
- Autonomous Control Architecture
- Conclusions

# Introduction

---



# Introduction

---

- Autonomous control architectures use models:
  - To reason about the system that control
  - The environment where they act
- They achieve a set of extended goals over a period of time, and are able to reason about faults with little or no human
- Given the initial state and external goals: they generate a set of actions
- If an action is not executed as expected: “recover”

# Outline

---

- Introduction
- **Autonomous architectures**
- Conclusions

# Classification

---

- Attending to the “**reactivity/deliberativeness**” of the system, the main architectural approaches are:
  - Reactive, procedural approach (non-deliberative)
    - Purely reactive controllers
    - Symbolic reactive controllers
  - Hybrid approach
    - Three-tiered controllers
    - Model-based, planning centric methods

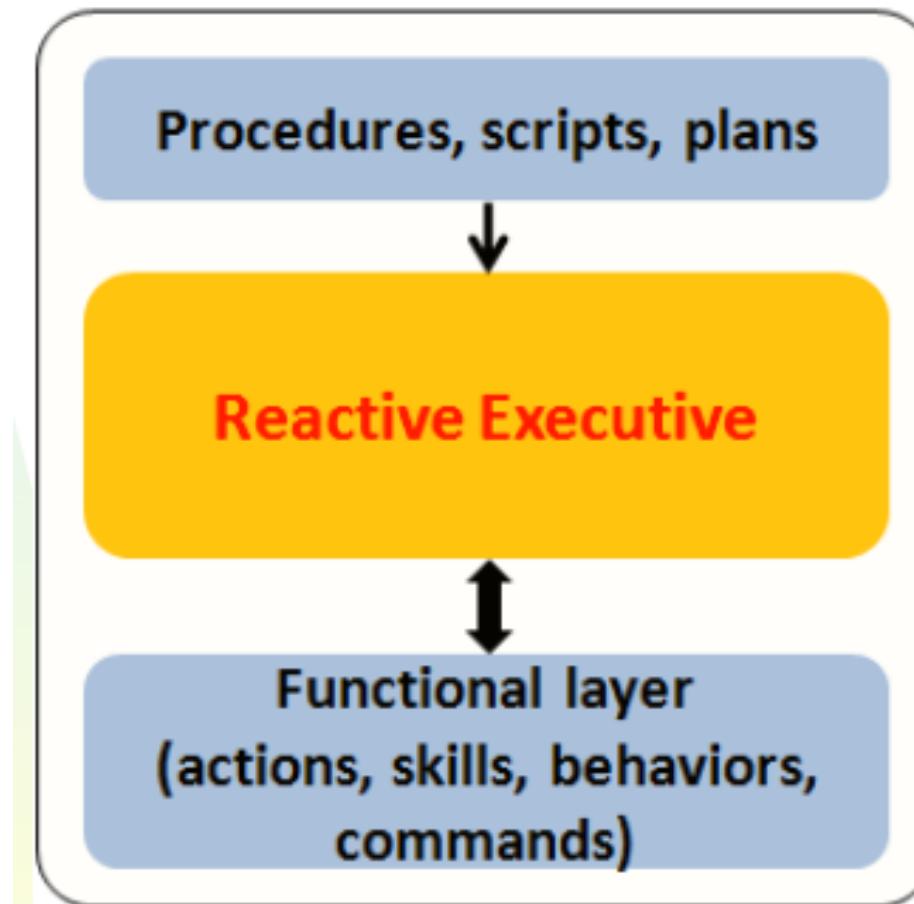
# Reactive (non-deliberative)

---

- **Purely reactive controllers**: motor action as response to the collected data by sensors without reasoning about them (no internal state is contained)
- **Symbolic reactive controllers**: an intermediate “decision-making” step that infers an action output from the sensor input and a symbolic representation is introduced (based on behaviours)
- Suitable when:
  - Real world cannot be accurately modeled
  - Uncertainty is quite delimited
  - Real-time warranty is a safety-critical concern

# Reactive (non-deliberative)

---



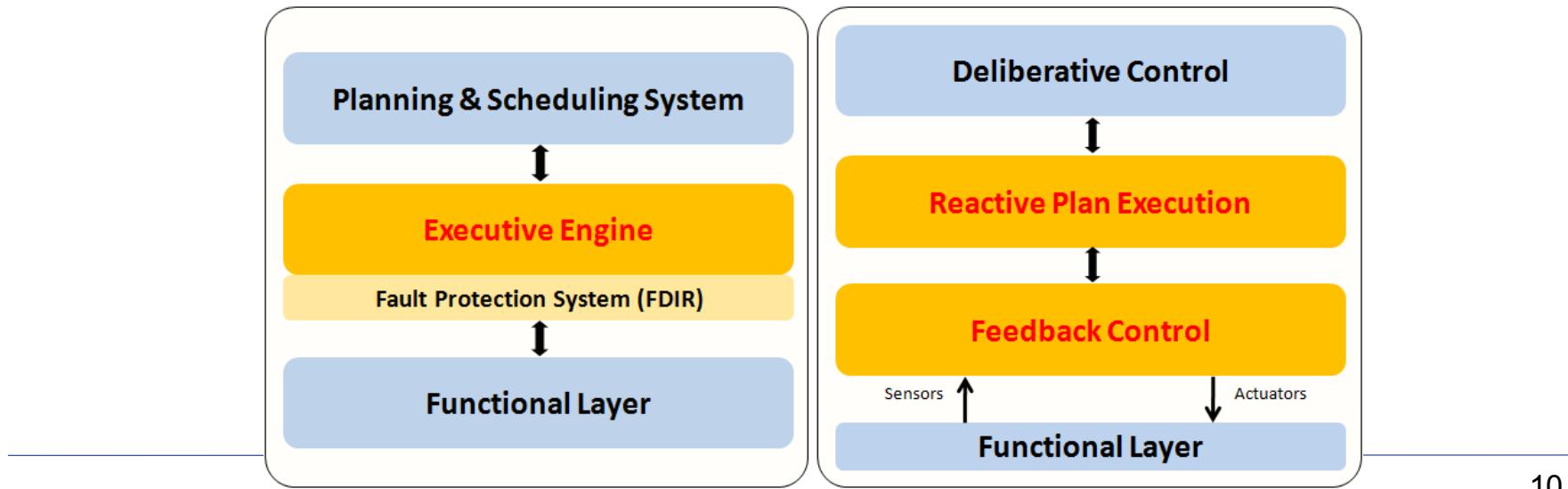
# Reactive (non-deliberative)

---

- Examples:
  - **Subsumption architecture** [Brooks, R.A., 1986]
    - System decomposed into concurrent (hierarchically arranged) behaviors: higher layers represent more abstract behaviors, and have lower priorities than the lowers
    - Behavior: sense-act function which maps perceptual inputs to actions
  - **Agent Network Architecture (ANA)** [Maes, P., 1991]
    - Similar system decomposition
    - Behaviors modules defined by their pre-conditions and post-conditions (interfaces), and by a (dynamic-valued) activation level

# Three layer architectures (hybrid)

- Interleave a planning step using a model of the world (implements Sense-Plan-Act cycle)
- Limitations:
  - Monolithic planning cycle
  - Slow reactivity when deliberation is needed
  - Hard scalability and robustness decreasing



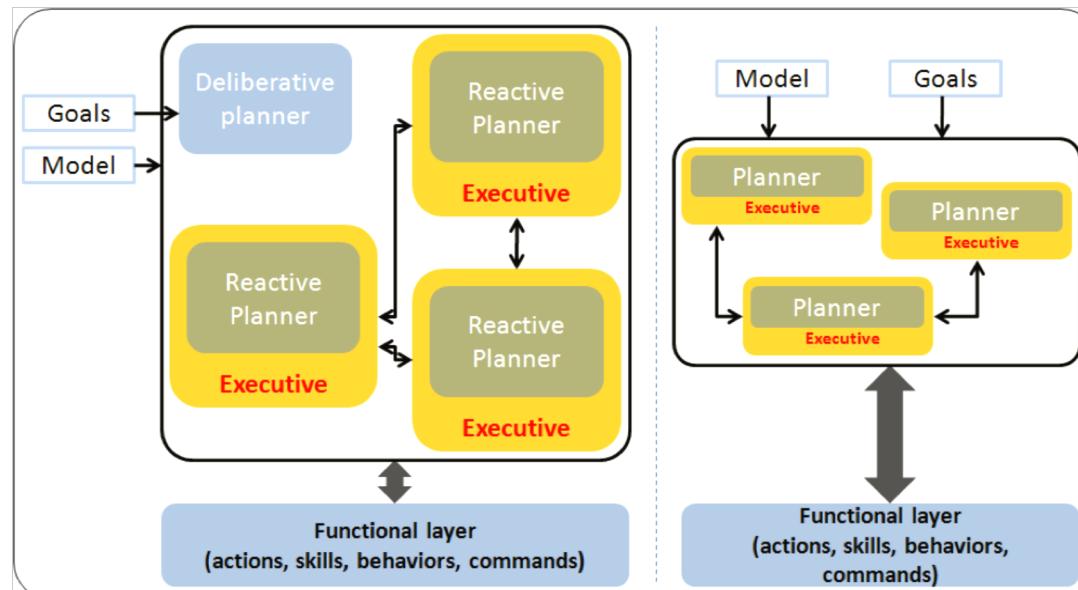
# Three layer architectures (hybrid)

---

- Examples:
  - **3T** [Bonasso et al., 1997]
    - 3 layers
  - **Tripodal Control Architecture** [Kim, G. and Chung, W., 2006]
    - Consist of a deliberative, a sequencing and a reactive layer.
    - The deliberative layer is the interface with the user and with the planning process execution
  - **ATLANTIS** [Gat, E., 1992]
    - Consists of a reactive controller, a sequencer and a deliberator
    - Includes monitor capabilities to re-planning/plan-repairing

# Model-based, planning centric methods (hybrid)

- Exploits automated planning and model-based reasoning at the core
- Divide-and-conquer scheme: system functionality is distributed
- Theoretically meets:
  - The efficiency/robustness of the reactive controllers
  - The high-level reasoning capabilities of the (purely) model-based approach



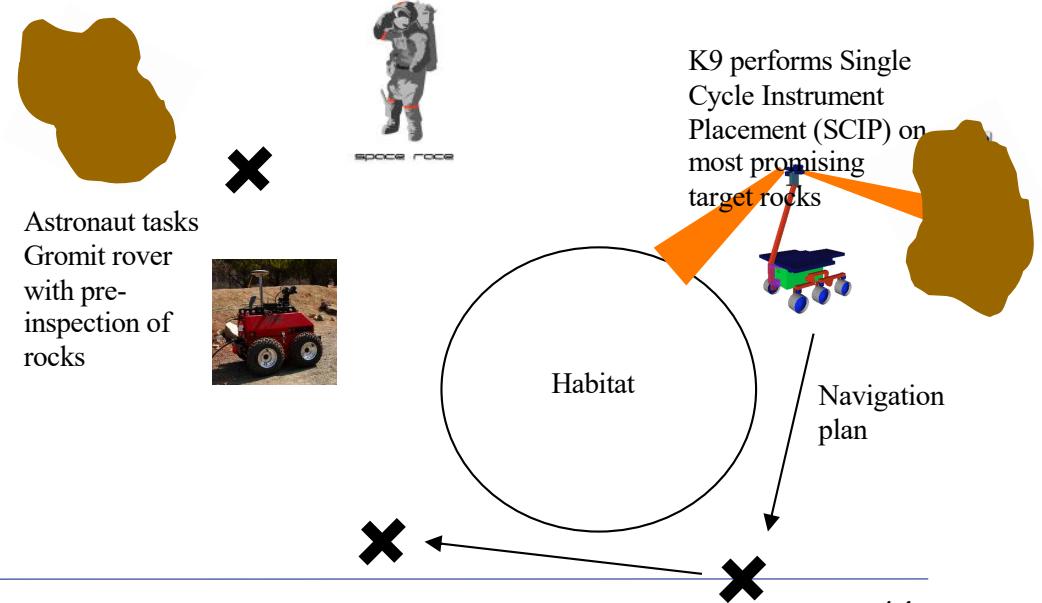
# An example: IDEA

---

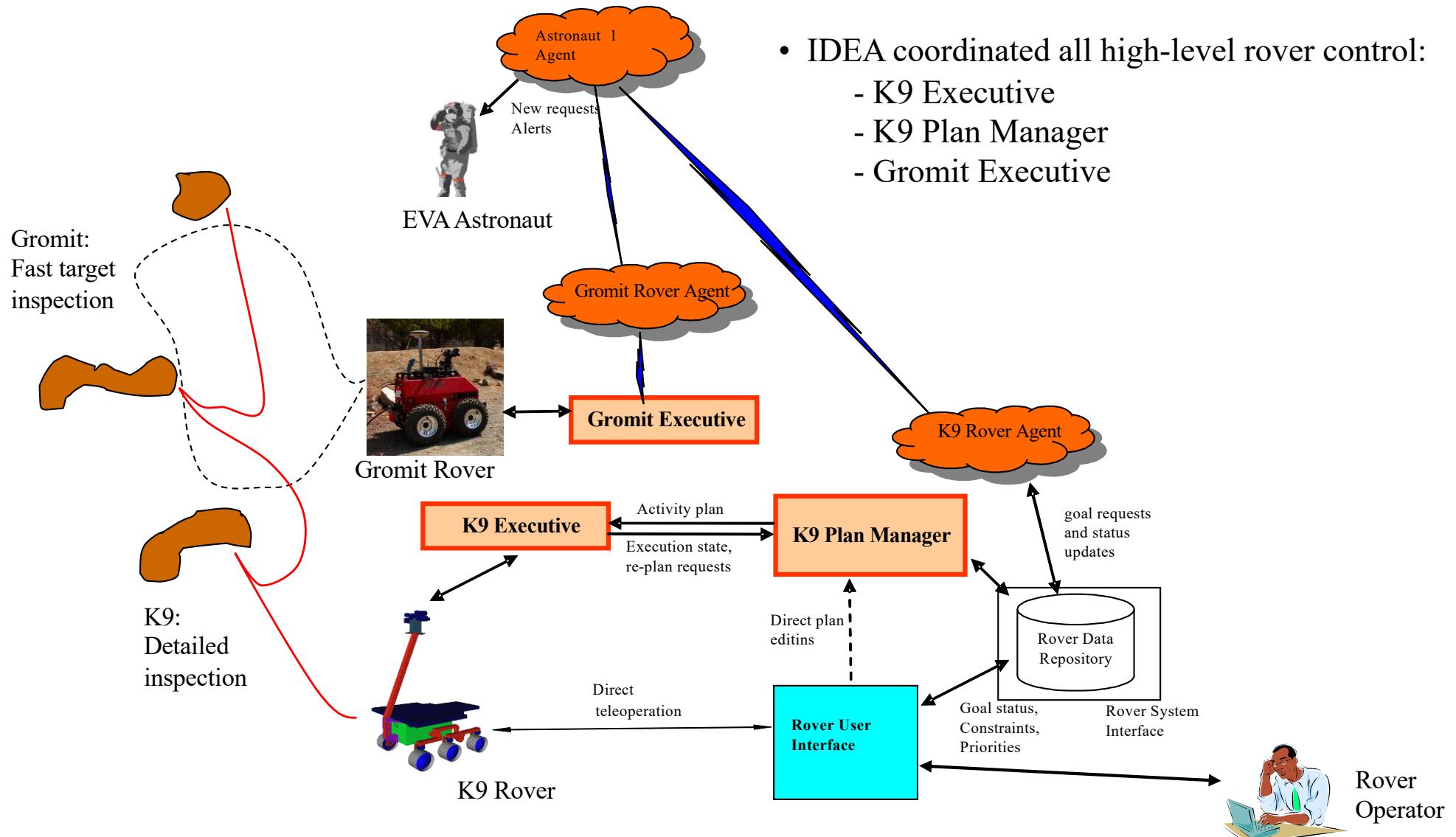
- IDEA is an architecture that enables high-level control by tightly integrating planning and execution
- Interleaved planning and execution
  - AI planning and scheduling is the core reasoning approach for execution
  - Planning over different horizons enables deliberation and reaction in a single framework
- Model-based control
  - Declarative models define system behavior and interactions with other systems
  - Models restrict plan search to legal behavior

# An example: IDEA

- Collaborative Decision Systems (CDS)
- Human-robot collaboration for planetary surface operations
  - Gromit rover (rock pre-inspection)
  - K9 Rover (detailed science sampling)
  - EVA astronaut (directing geology)
  - RoverOperator (oversees rover ops)
- Project Goals:
  - Demonstrate coordinated human-robot surface operations near human habitat
  - Coordinated field geology
  - Tightly integrate and test existing technologies
- IDEA K9 Exec and PlanManager
  - PlanManager: turns goal requests from RoverOperator into plans for K9
  - K9 Exec: using plans, oversees mobility, target tracking, instrument placement and measurements
- IDEA Gromit Exec
  - Coordinated mobility and panoramic image taking
- Results Synopsis
  - Successful demonstration of coordinated operations
  - IDEA greatly eased system integration



# CDS System Architecture

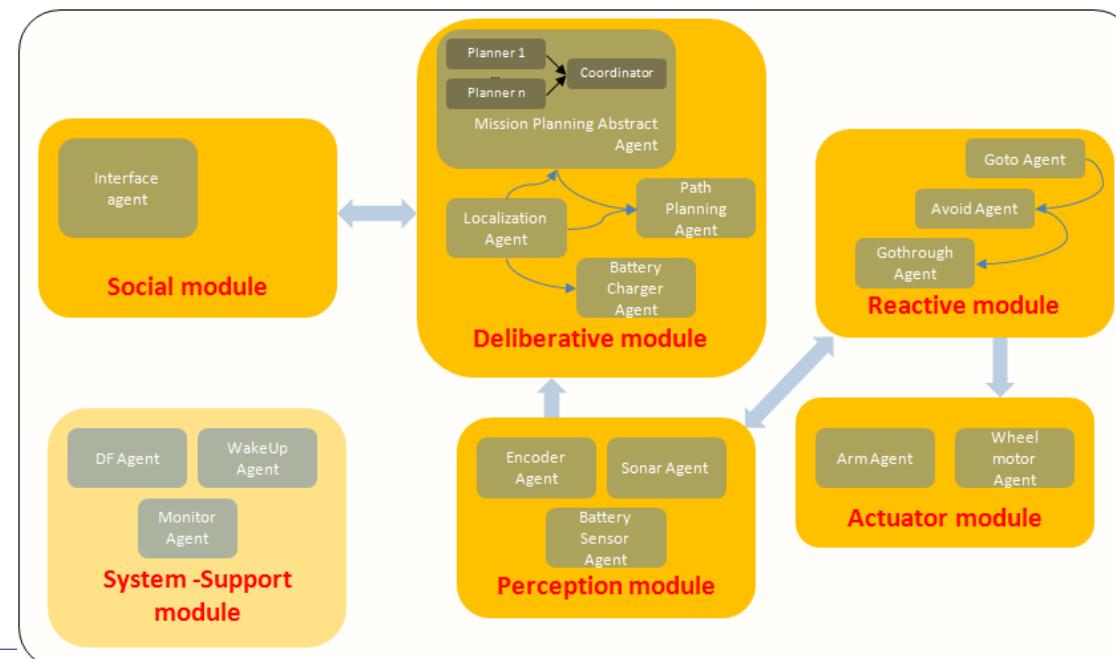


# Model-based, planning centric methods (hybrid)

## □ Examples

### ■ ARMADiCo [Innocenti, B., 2008]

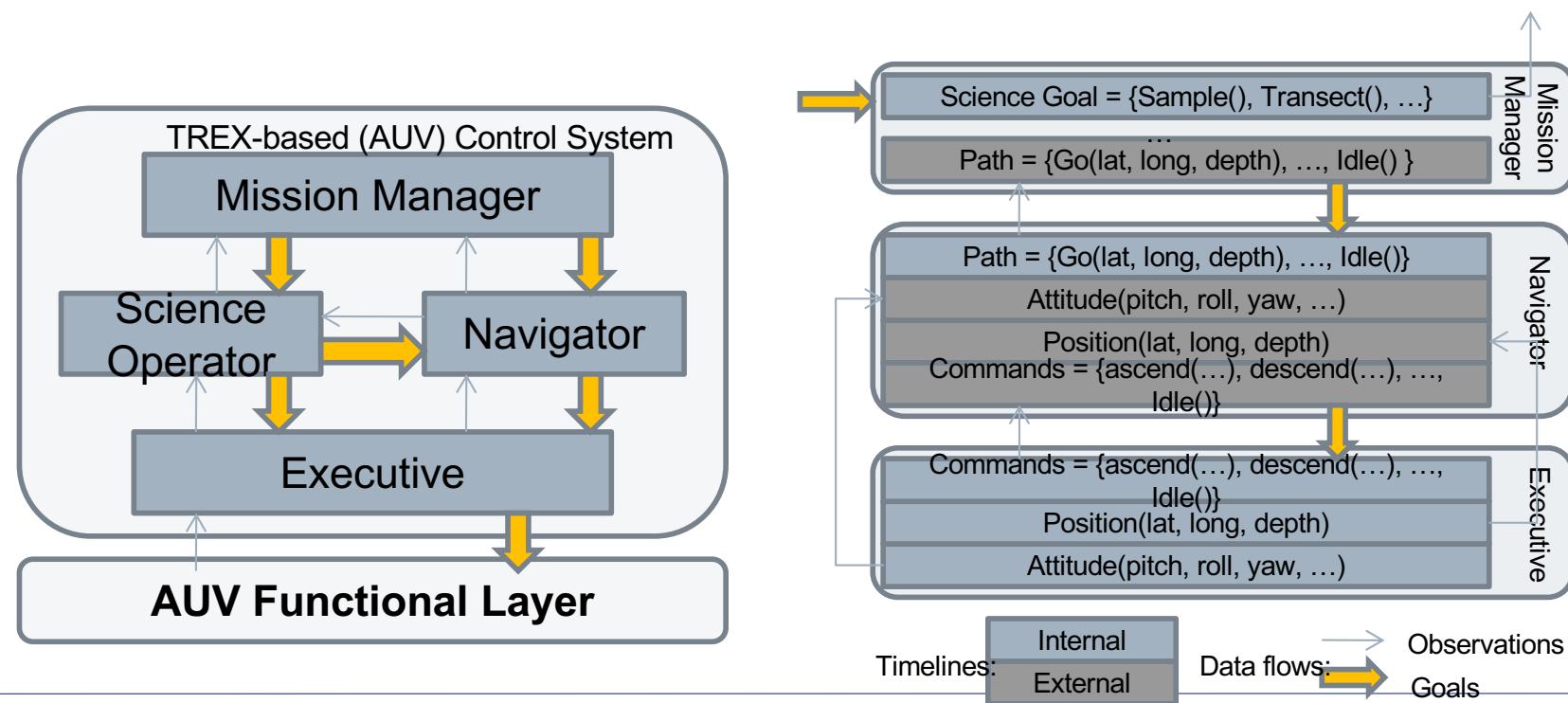
- Multi-agent and general purpose architecture for mobile robots
- Distributed communication scheme which allows agents compete for the available resources



# Model-based, planning centric methods (hybrid)

## □ Examples

- **TREX** [McGann, C. et al., 2008]
  - Combines goal- and event-driven behavior in a framework based on temporal reasoning & planning
  - Developed for Autonomous Underwater Vehicles



# Outline

---

- Introduction
- Autonomous Control Architecture
- Conclusions

# Conclusions

---

## □ What is the desired autonomy level?

Level	Description	Functions
E1	Mission execution under ground control; limited onboard capability for safety issues	Real-time control for nominal operations. Execution of time-tagged commands for safety issues
E2	Execution of pre-planned, ground-defined, mission operations onboard	Capability to store time-based commands in an on-board scheduler
E3	Execution of adaptive mission operations onboard	Event-based autonomous operations. Execution of on-board operations control procedures
E4	Execution of goal-oriented mission operations onboard	Goal-oriented mission re-planning

# Conclusions

---

- Is the scope the same in Planetary Exploration and Orbit?
  - ◆ In Orbital missions (Earth or Space Observations) are different from Planetary Exploration
  - ◆ Why? Surface on the planets present special challenges for autonomous robots
  - ◆ The robot must navigate in these environments avoiding obstacles where it can collide or fall in a hole

# Conclusions

---

## □ Where can we apply AI in robotic systems?

- ◆ Diagnosis and reconfiguration
- ◆ Navigation and Path Planning (depends on the mission)
- ◆ Planning/Scheduling and Intelligent Execution
- ◆ Command sequence generation
- ◆ Data processing and validation
- ◆ Vision

# Conclusions

---

- A system is autonomous if it is able to achieve their goals with little or no human supervision
- Traditional systems are based on 3T-arquitectures
- New approaches distribute the systems between different controller



Universidad  
de Alcalá

# Autonomous Control Architectures

---

