Agents and Environments
Lecture 3

Jim Rehg

January 15, 2016

College of Computing
Georgia Tech

(Slides based in part on slides from Andrea Thomaz)

Task Setting

The problem the agent is trying to solve:

P – Performance Metric

E – Environment

A – Actuators

S – Sensors

Automated Taxi Problem (coming soon…)

P – Safe, fast, legal, max revenue, min cost, min fuel, …

E – city roads, traffic, pedestrians, bikers, construction, …

A – Car controls (steering, gas pedal) and human interface

S – Cameras, radar, laser rangefinder, GPS, mapping, engine sensors, human input devices

Environment Types

Fully Observable	Partially Observable
Deterministic	Stochastic
Episodic	Sequential
Static	Dynamic
Discrete	Continuous
Single-Agent	Multi-Agent

Summary

Agents map states to actions

States and actions represented as tuples

Agents should be rational

Select the action that maximizes the outcome

Task defined by PEAS

Performance metric, Environment, Actuation, Sensors

Environment Types influence the difficulty of the problem

Agent Program Design

Sensors

Just build a big table?

1 row for every possible state *s* in the set *S* of all states

Each row contains the best action for that state

Totally impractical!

State (perception) <Ball-loc, Sq-loc, Hand-loc>	Action <L, R, Pick(obj), Drop>
Ball-A, Sq-Hand, Hand-A	R
Ball-A, Sq-Hand, Hand-B	Drop
Ball-A, Sq-B, Hand-B	L
Ball-A, Sq-B, Hand-A	Pick(ball)
Ball-Hand, Sq-B, Hand-A	R
Ball-Hand, Sq-B, Hand-B	Drop
Ball-B, Sq-B, Hand-B	None
...	

Actuators

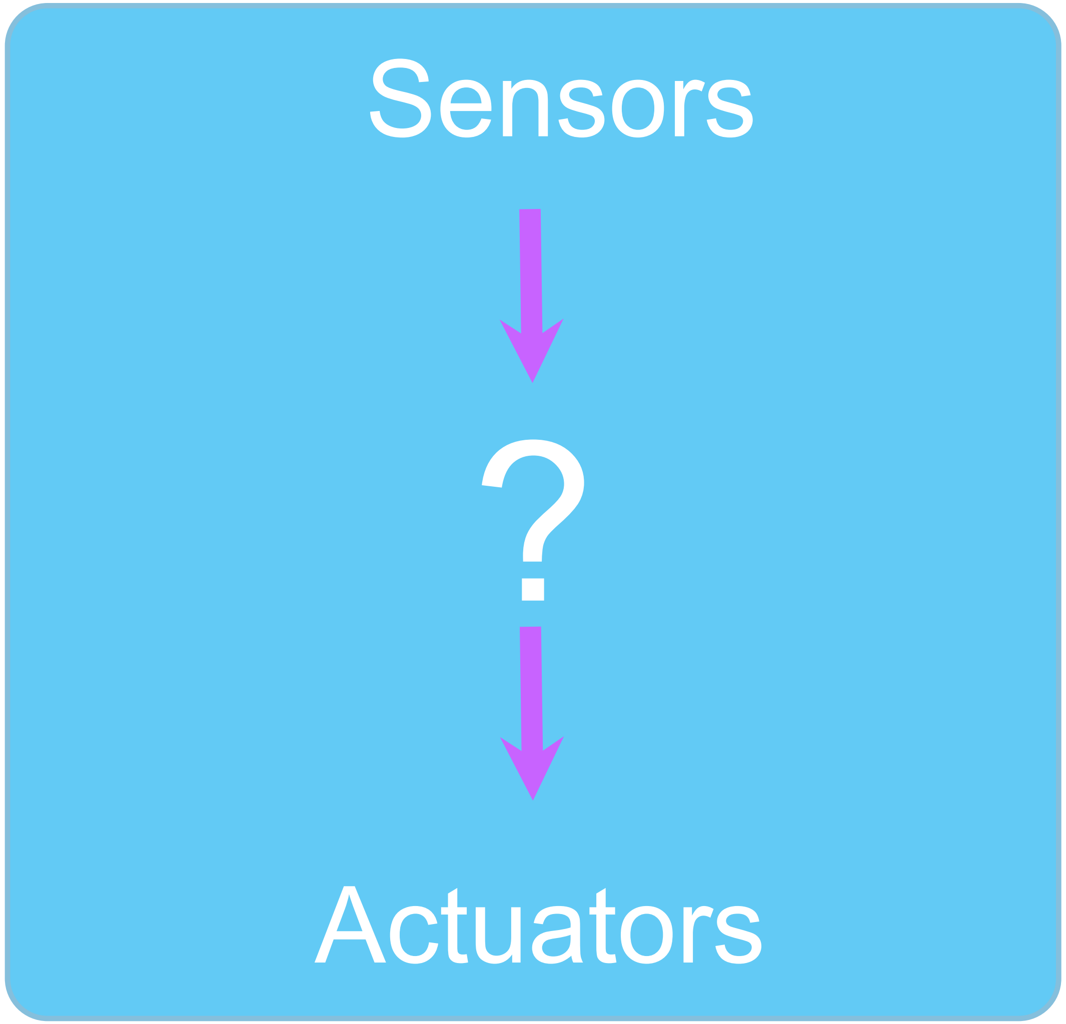
Types of Agents

Reflex

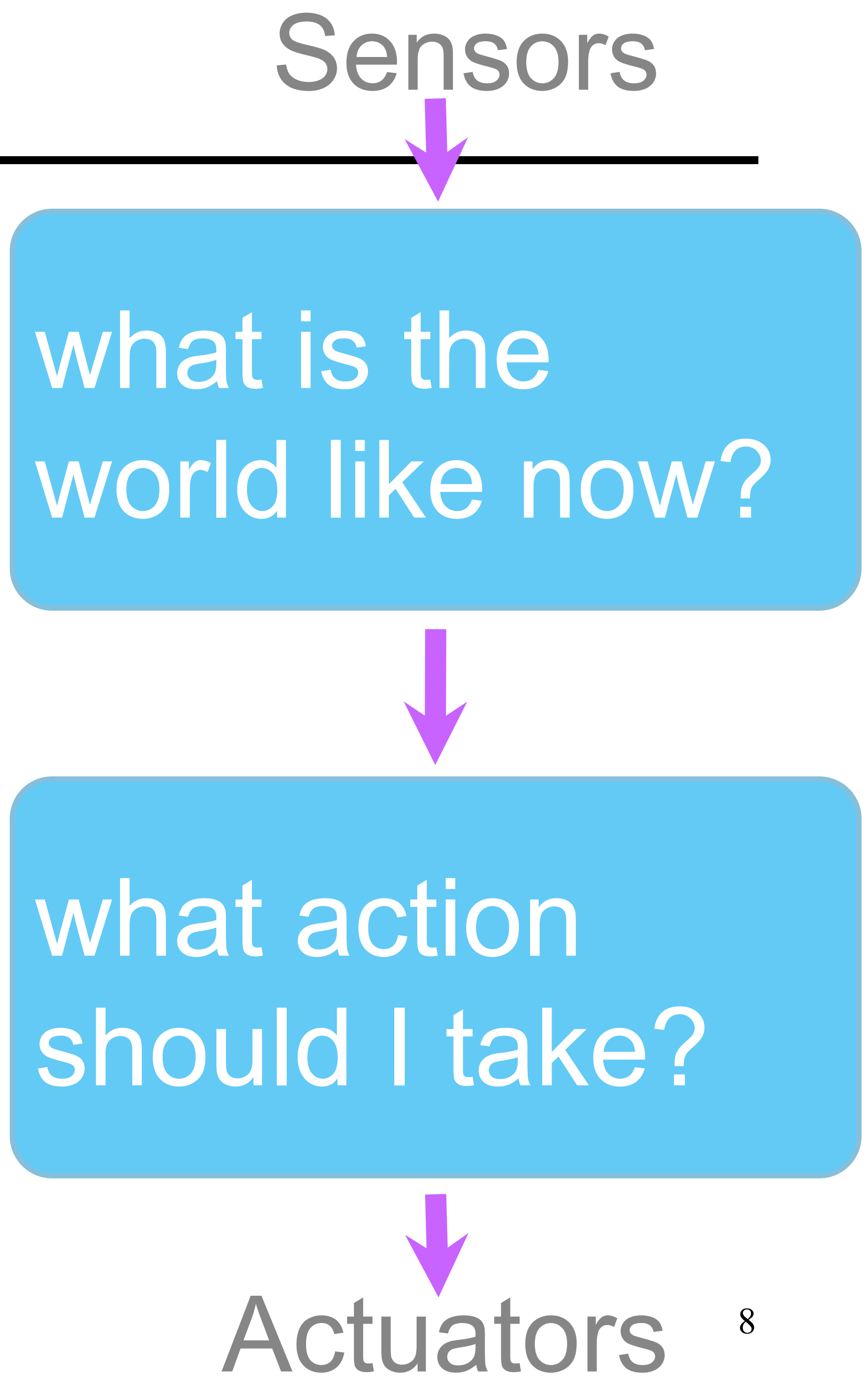
Model-based

Goal-based

Utility-based



Reflex Agent

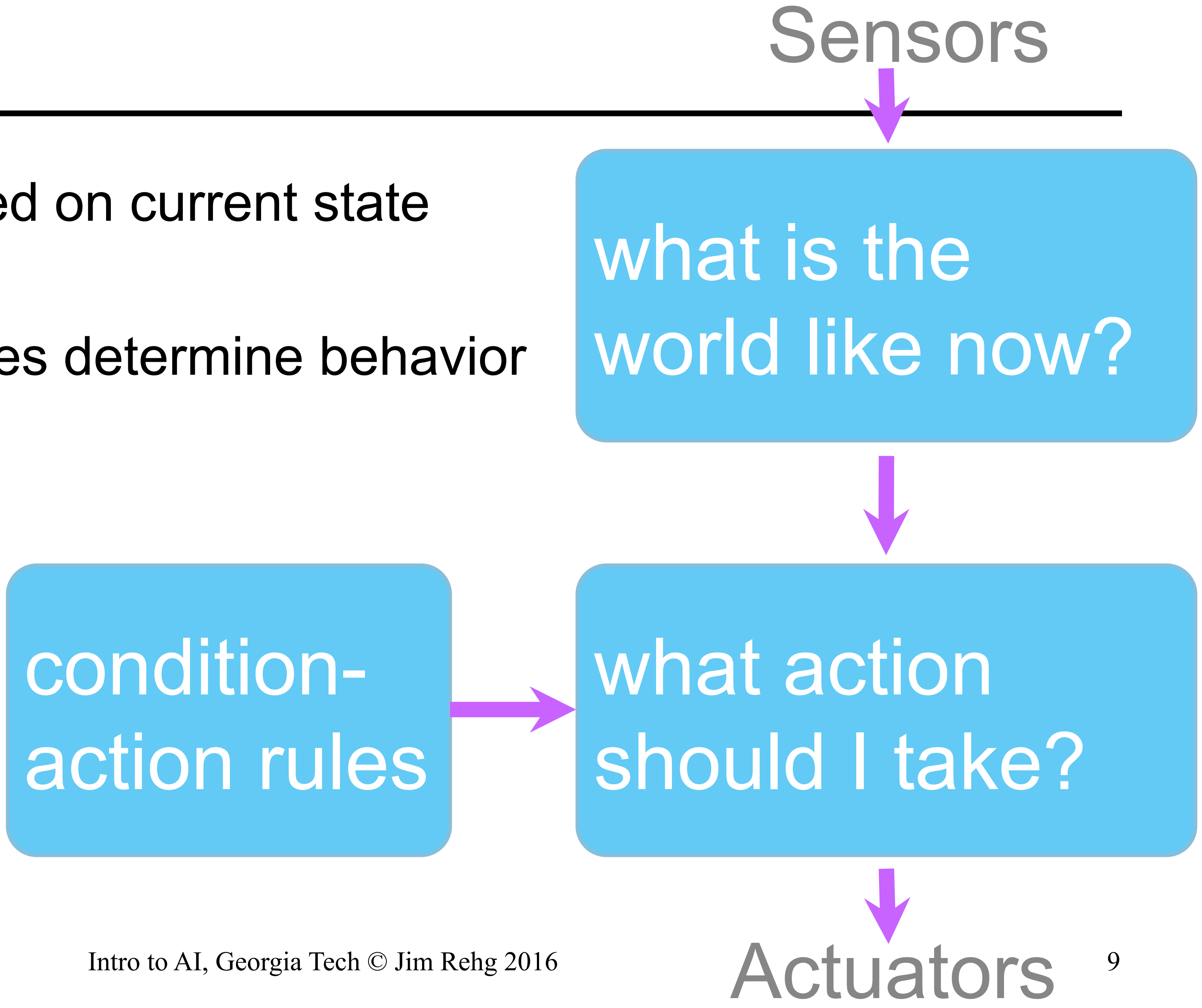


Reflex Agent

Perform action based on current state

Condition-action rules determine behavior

Better than a table?



Condition-Action Tables vs Rules

State (perception) <Ball-loc, Sq-loc, Hand-loc>	Action <L, R, Pick(obj), Drop>
A, A, A	Pickup (Square)
B, A, A	Pickup (Square)
…	

vs.

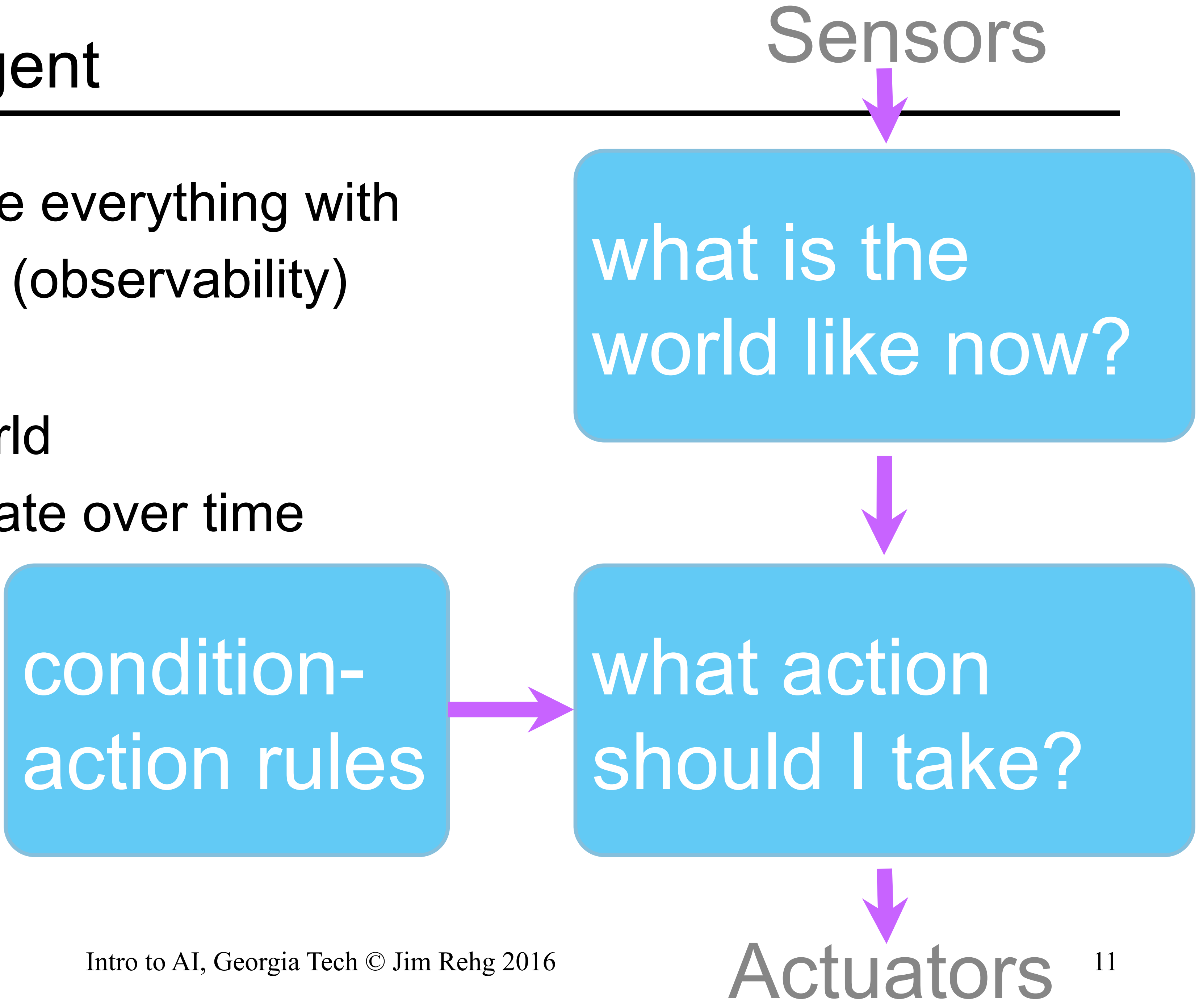
IF: (sq-A ^ Hand-A ^ !Ball-hand)
THEN: Pickup (Square)

Model-Based Agent

What if you can't see everything with current percepts? (observability)

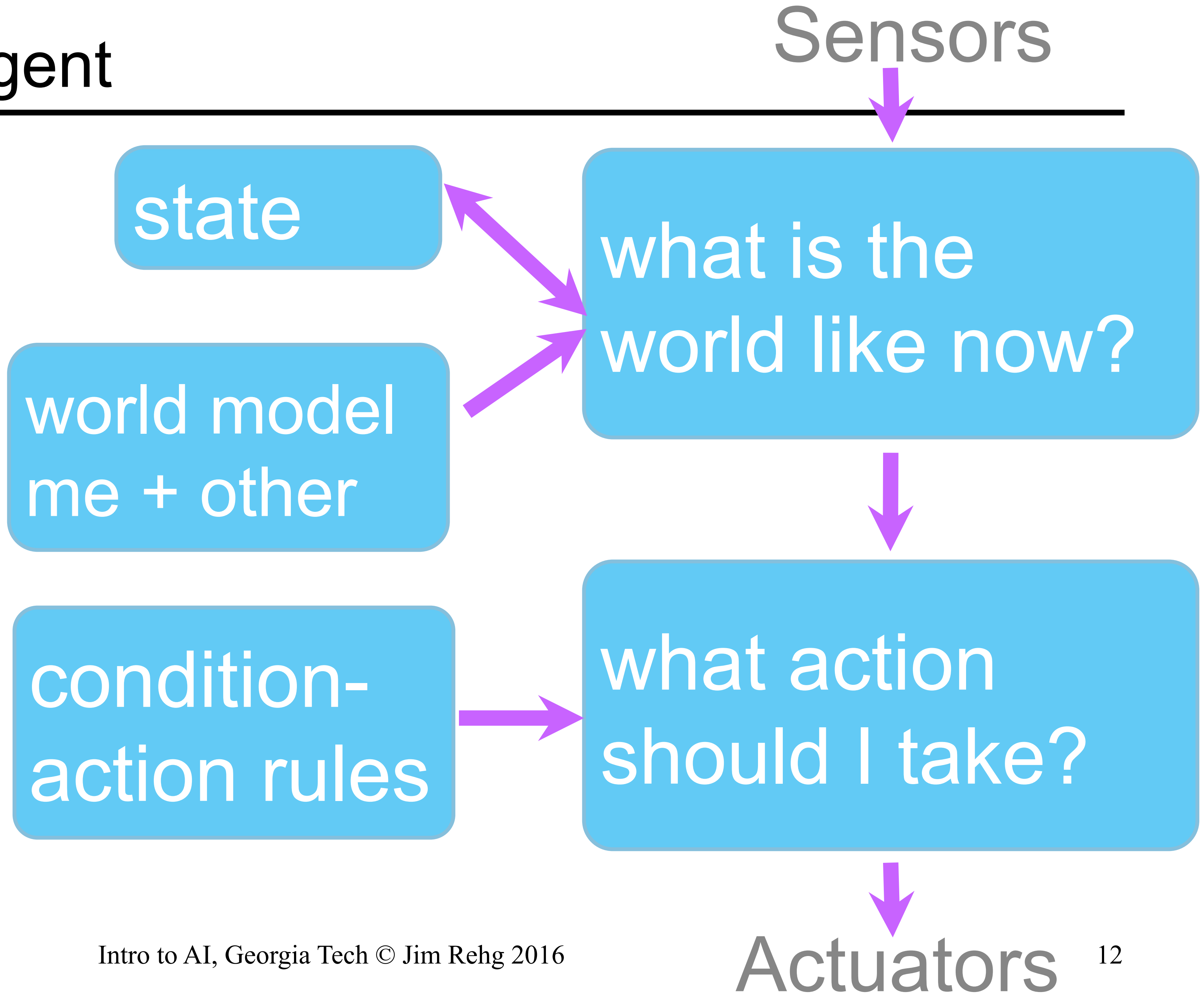
Create model of world

Track changes in state over time



Model-Based Agent

Change “observer” to include prior world states

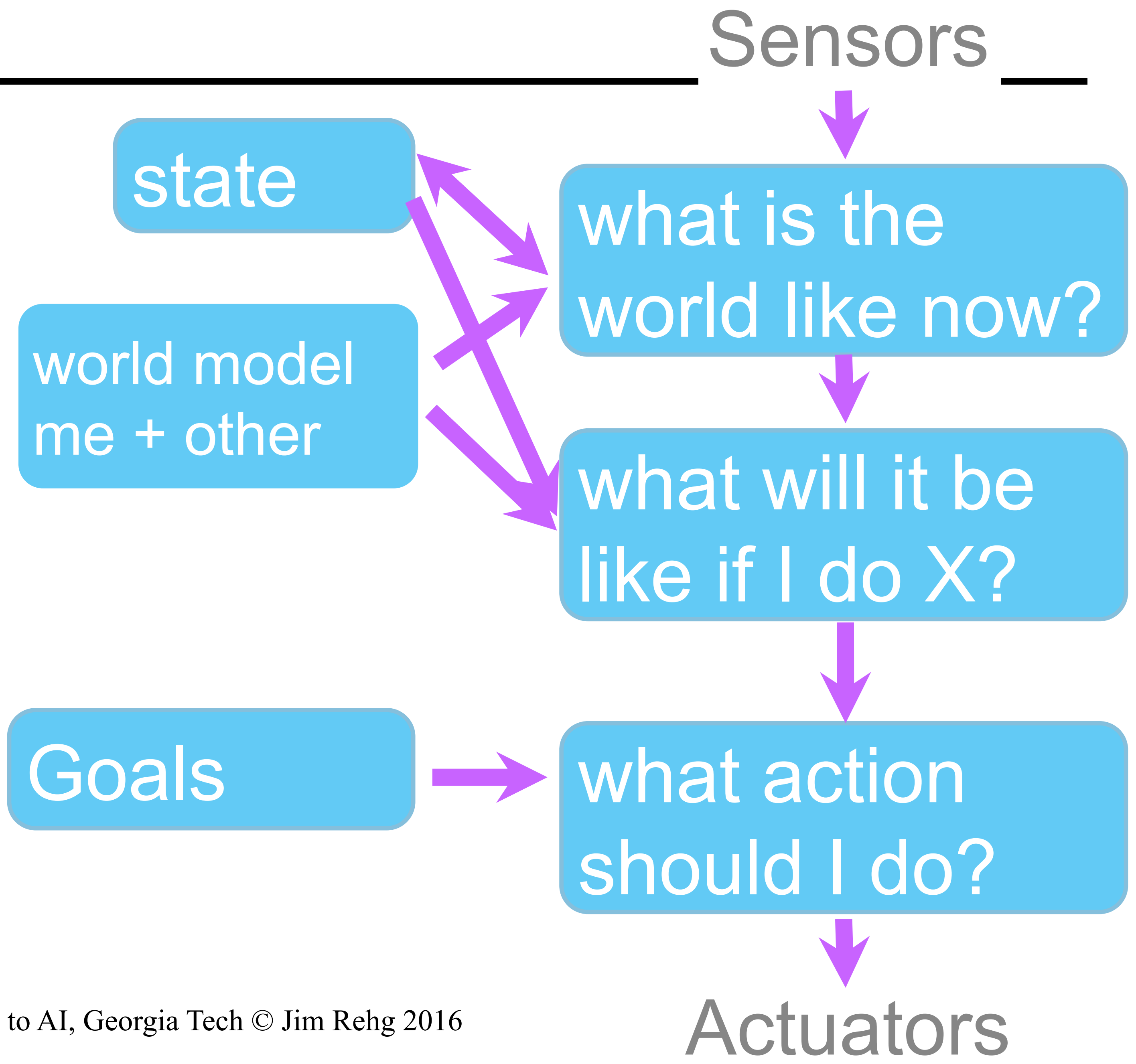


Goal-Based Agent

Action not only
Action not only
dependent on
state history

Add planning

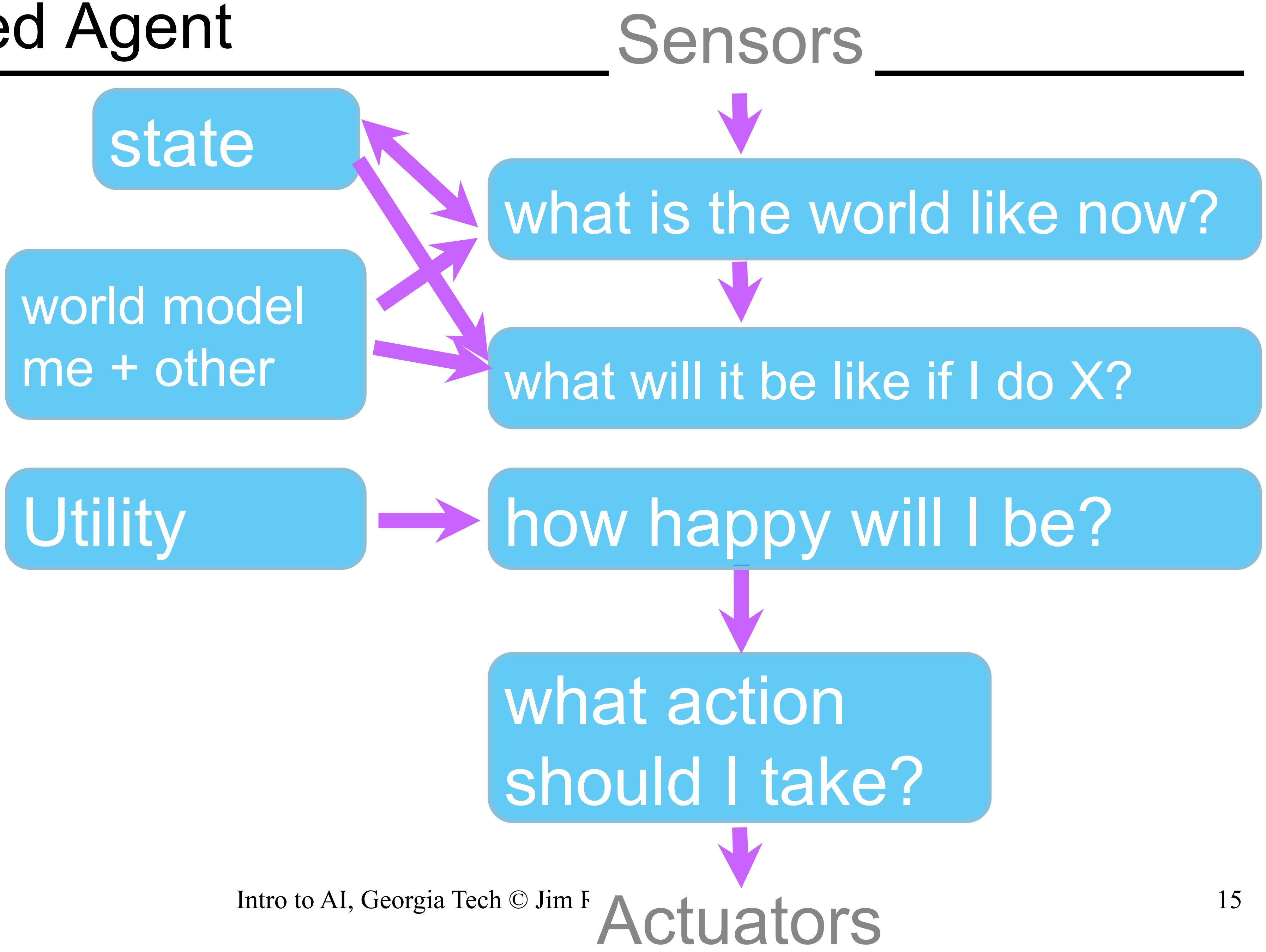
Look ahead to
predict future
state



Utility-Based Agent

We need to provide additional information to select the best path

Utility-Based Agent



How to Does an Agent Obtain a Model?

Model is “programmed” by humans

Knowledge Engineering

- Query Experts
- Build Knowledge Base
- Interface to Agent

Cycorp®

Started in 1984 by Doug Lenat

Other examples?

How to Does an Agent Obtain a Model?

Model is “programmed” by humans

Knowledge Engineering

- Query Experts
- Build Knowledge Base
- Interface to Agent

Cycorp

Started in 1984 by Doug Lenat

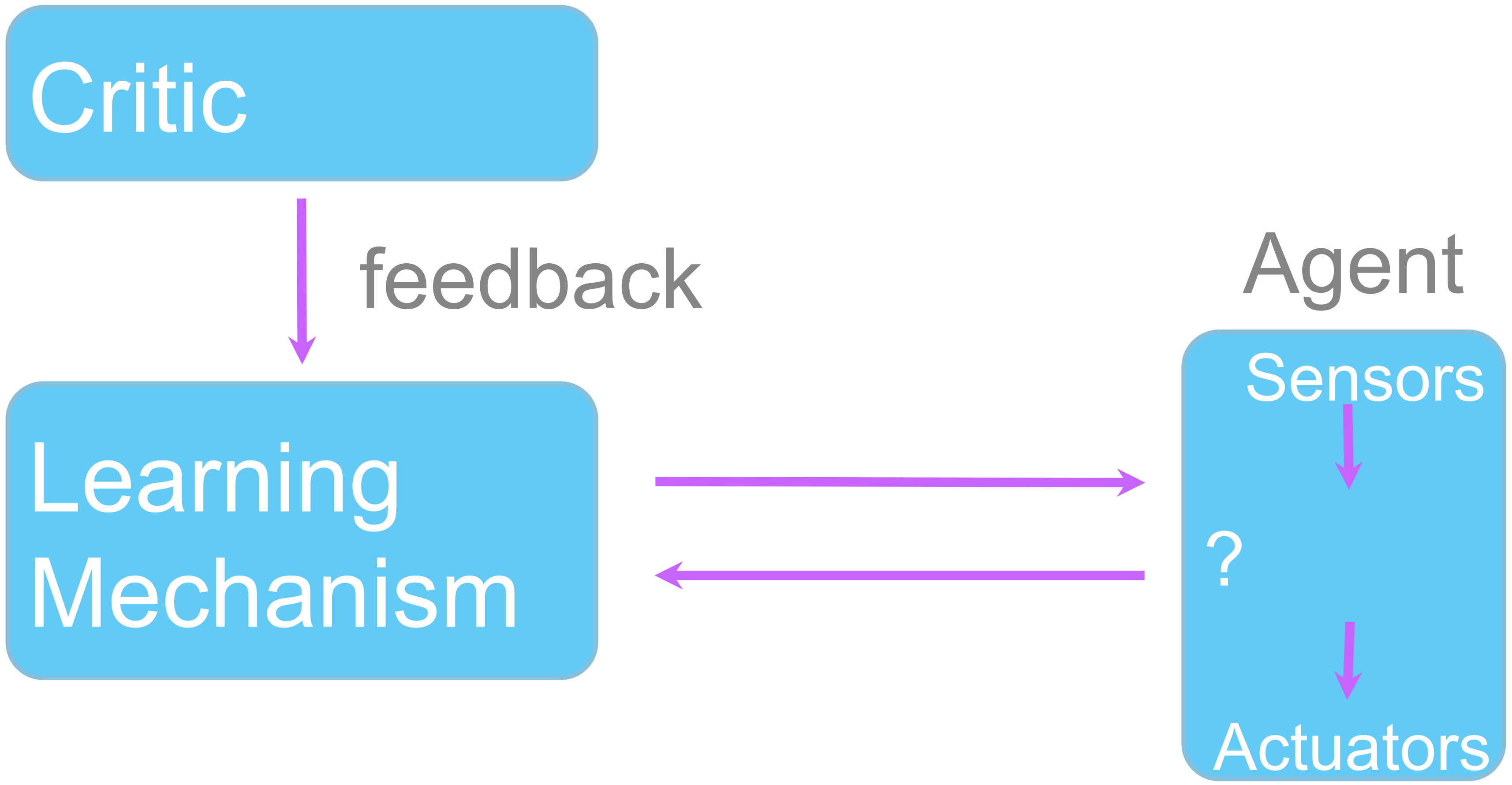
Other examples?

Agent learns a model

From labeled examples (supervised)

From its own experience (unsupervised)

Learning Agents



Many things to learn: World Model, Action Model, Utility Function, etc.

Google DeepMind – Deep Q Learning for Atari Games

V. Minh, et. al. “Human-level control through deep reinforcement learning”
Nature 518, pp. 529–533 (26 February 2015) doi:10.1038/nature14236

Google DeepMind

Deep Q-Learning

- *Deep* = General function approximation technology for learning input-output mappings
- *Q-Learning* = A simple form of reinforcement learning, where you learn to select actions optimally over time

Mastered about half of the classic Atari 2600 games

- Space Invaders, Pong, etc. (but not Pac Man)

Company co-founded by Demis Hassabis, acquired by Google in 2014 for a reported $650M

Questions?