Administrative Issues

- Please send an email to qtipu@usc.edu so that you can be on the class mailing list: csci561@yahoogroups.com

- Please when sending homework-related emails use
  Subject: HW: question about …

- Quamrul Tipu: Office hours: Fridays, 2-4pm, SAL-211
- Seokkyung Sung: Office hours: Weds, 10am-12pm, SAL-229

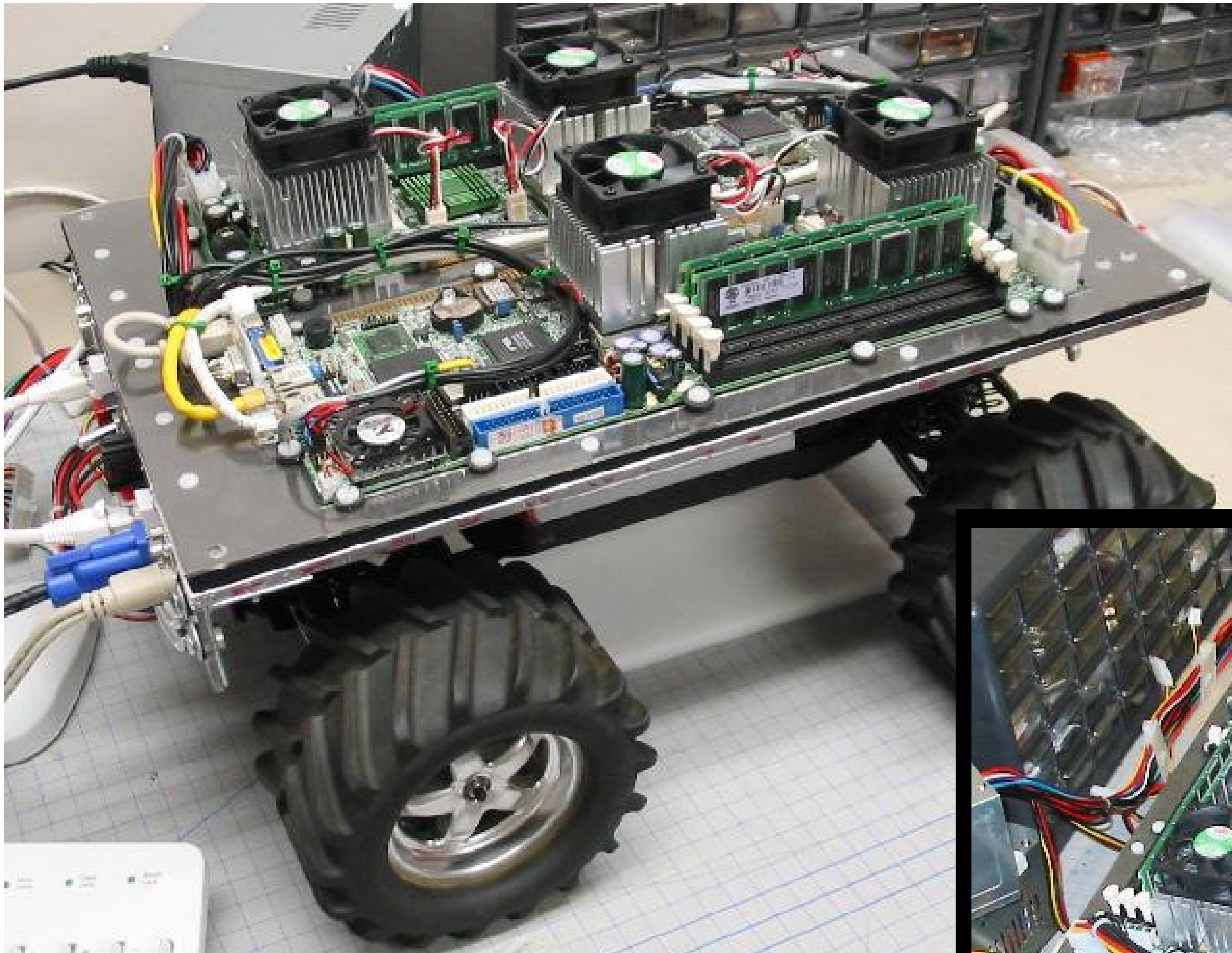
- Web page: http://iLab.usc.edu (and follow the links)
  http://www-scf.usc.edu/~csci561a/

Last time: A driving example: Beobots

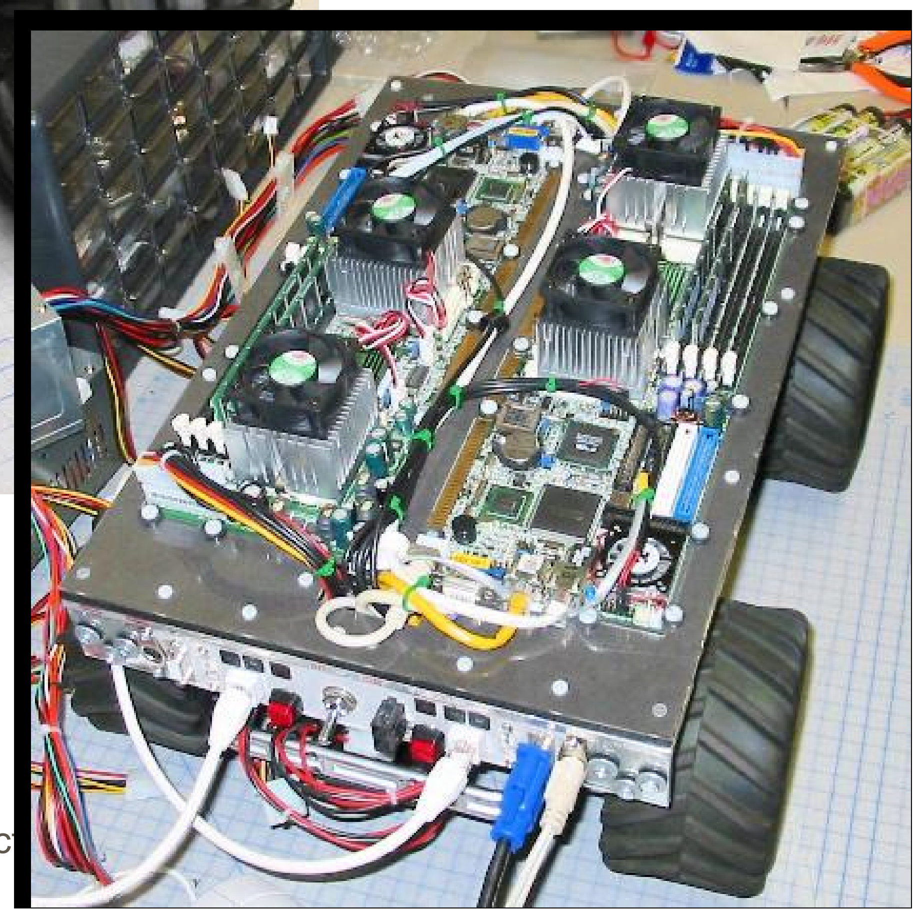
- **Goal:** build robots that can operate in unconstrained environments and that can solve a wide variety of tasks.

- **We have:**
  - Lots of CPU power
  - Prototype robotics platform
  - Visual system to find interesting objects in the world
  - Visual system to recognize/identify some of these objects
  - Visual system to know the type of scenery the robot is in

- **We need to:**
  - Build an internal representation of the world
  - Understand what the user wants
  - Act upon user requests / solve user problems



Beowulf + Robot =
"Beobot"



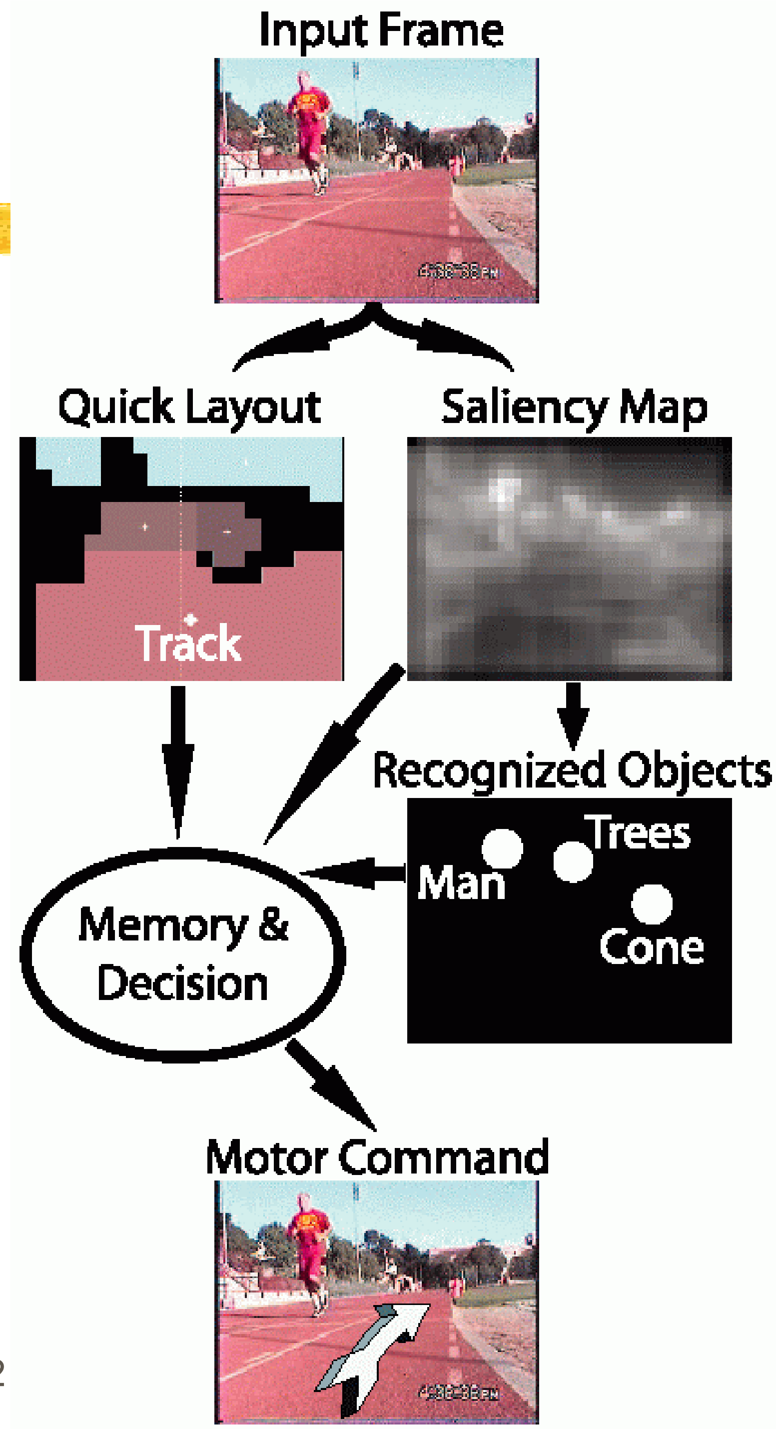
Prototype

Stripped-down version of proposed general system, for simplified goal: drive around USC olympic track, avoiding obstacles

Operates at 30fps on quad-CPU Beobot;

Layout & saliency very robust;

Object recognition often confused by background clutter.



Major issues

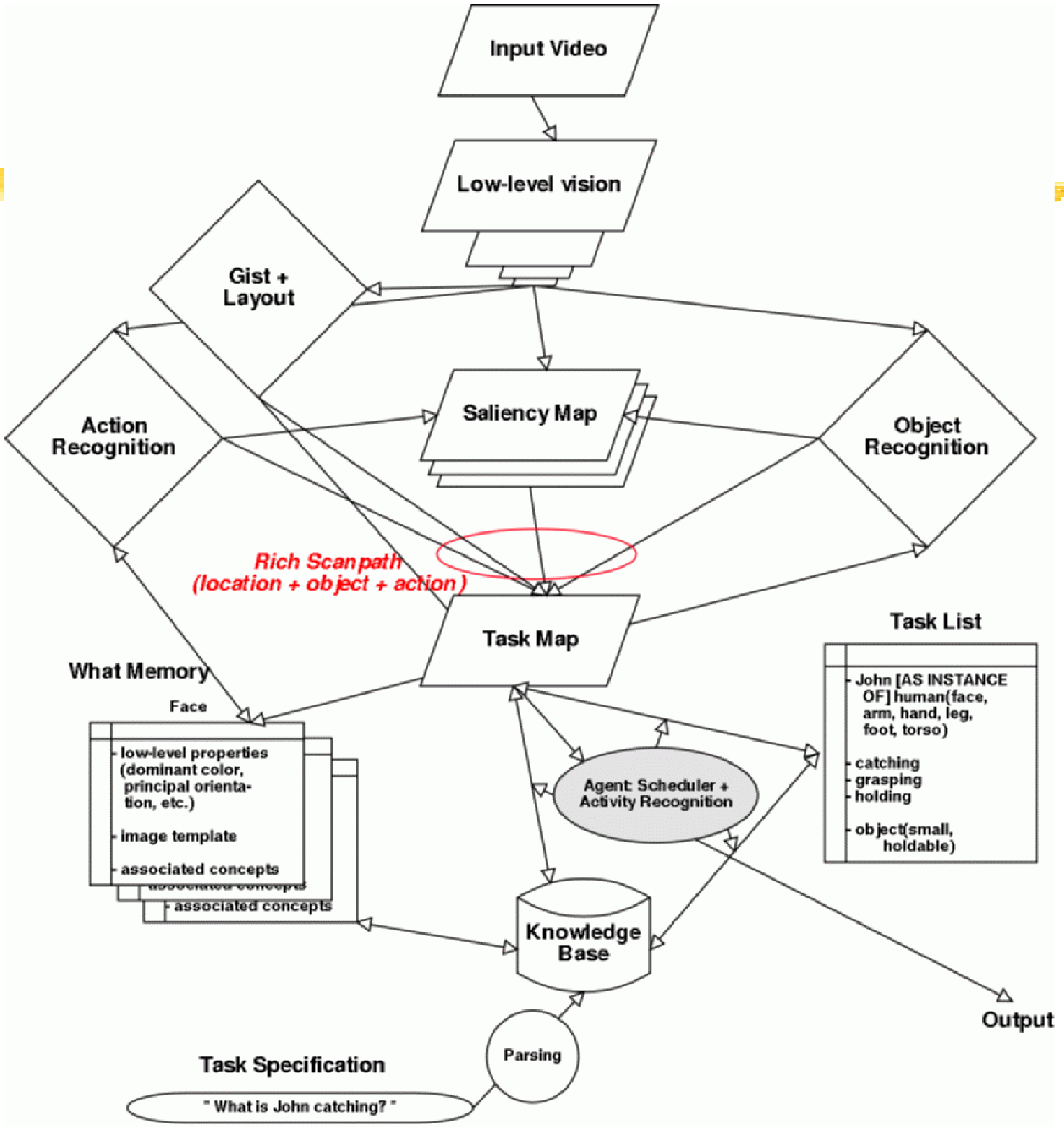
- How to represent knowledge about the world?

- How to react to new perceived events?
- How to integrate new percepts to past experience?

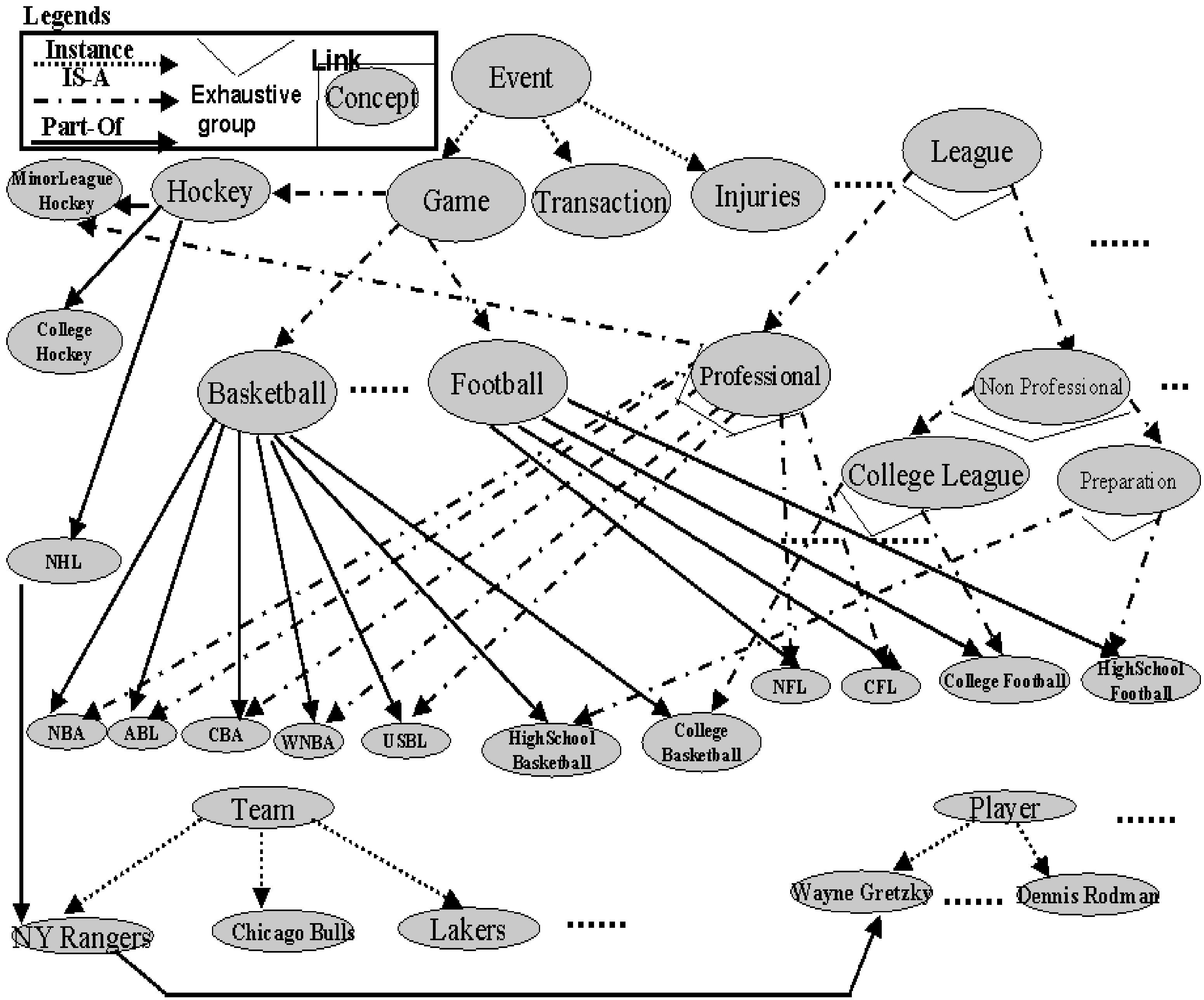
- How to understand the user?
- How to optimize balance between user goals & environment constraints?
- How to use reasoning to decide on the best course of action?
- How to communicate back with the user?

- How to plan ahead?
- How to learn from experience?

General architecture



Ontology

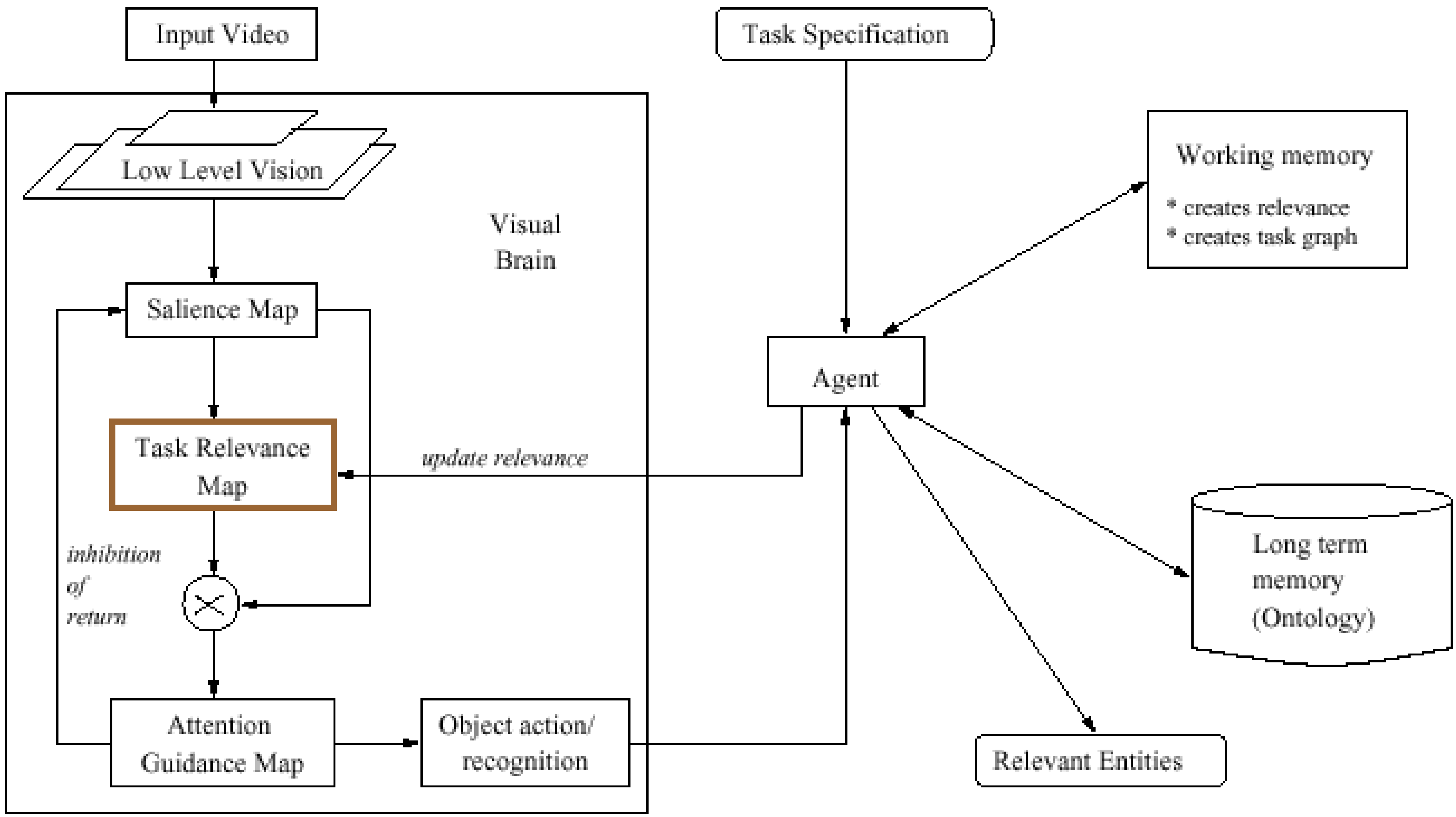


Khan & McLeod, 2000

The task-relevance map

Navalpakkam & Itti, BMCV'02

Scalar topographic map, with higher values at more relevant locations



More formally: how do we do it?

- Use ontology to describe categories, objects and relationships:

  Either with unary predicates, e.g., Human(John),

  Or with reified categories, e.g., John $\in$ Humans,

  And with rules that express relationships or properties,

  e.g., $\forall x$ Human(x) $\Rightarrow$ SinglePiece(x) $\wedge$ Mobile(x) $\wedge$ Deformable(x)

- Use ontology to expand concepts to related concepts:

  E.g., parsing question yields "LookFor(catching)"

  Assume a category HandActions and a taxonomy defined by

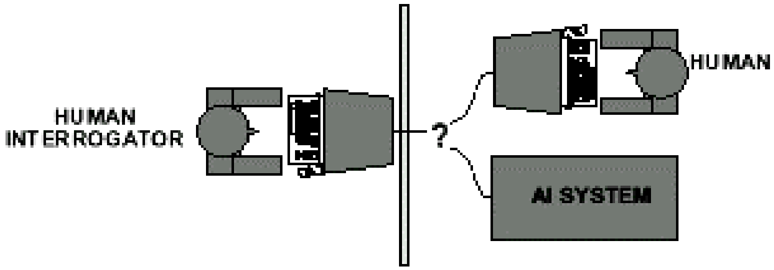
  catching $\in$ HandActions, grasping $\in$ HandActions, etc.

  We can expand "LookFor(catching)" to looking for other actions in the category where catching belongs through a simple expansion rule:

  $\forall a,b,c \quad a \in c \wedge b \in c \wedge \text{LookFor}(a) \Rightarrow \text{LookFor}(b)$

Last Time: Acting Humanly: The Full Turing Test

- Alan Turing's 1950 article *Computing Machinery and Intelligence* discussed conditions for considering a machine to be intelligent
  - "Can machines think?" ←→ "Can machines behave intelligently?"
  - The Turing test (The Imitation Game): Operational definition of intelligence.

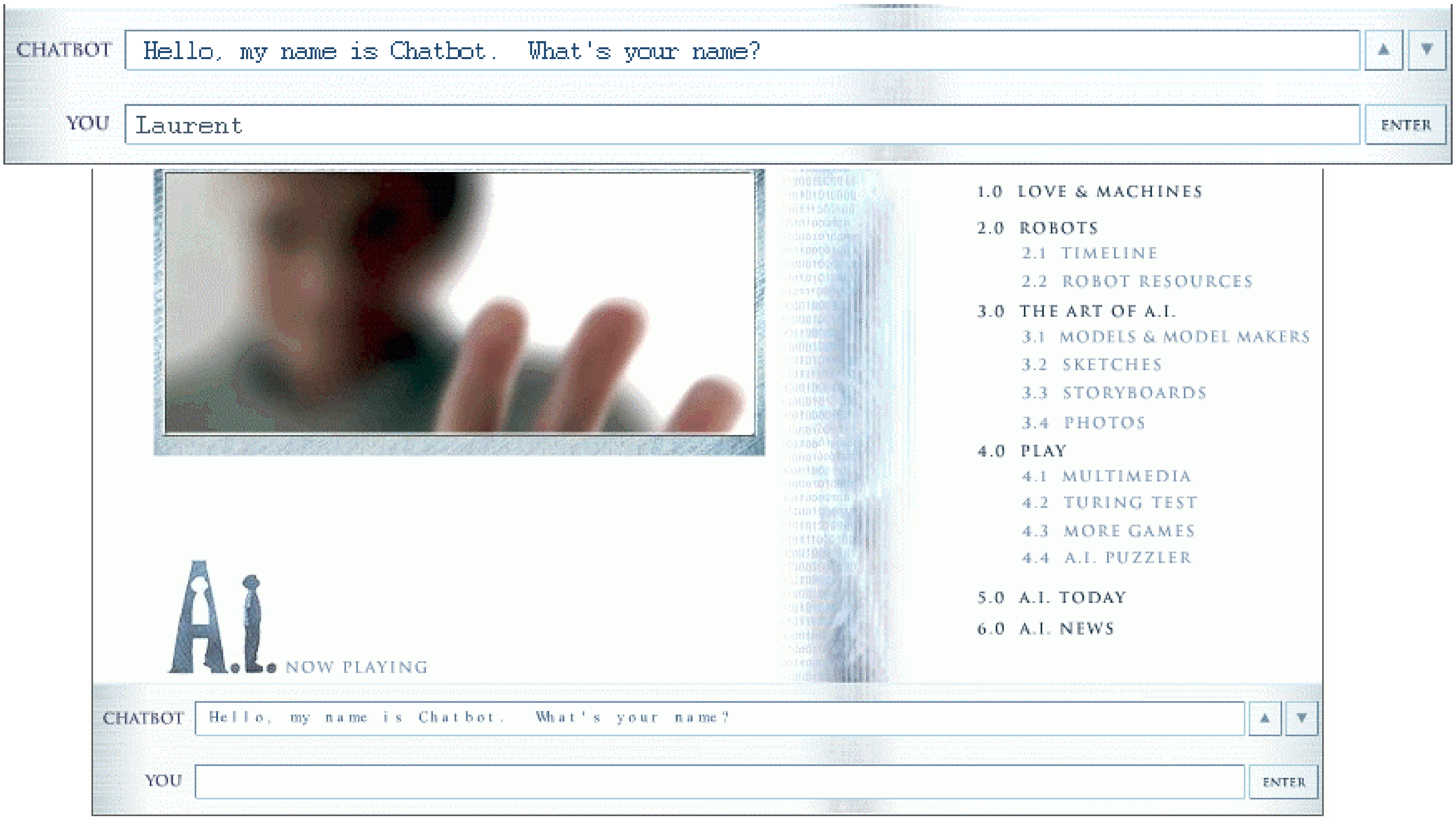


- Computer needs to posses: Natural language processing, Knowledge representation, Automated reasoning, and Machine learning
- Problem: 1) Turing test is not reproducible, constructive, and amenable to mathematic analysis. 2) What about physical interaction with interrogator and environment?
- Total Turing Test: Requires physical interaction and needs perception and actuation.

Last time: The Turing Test

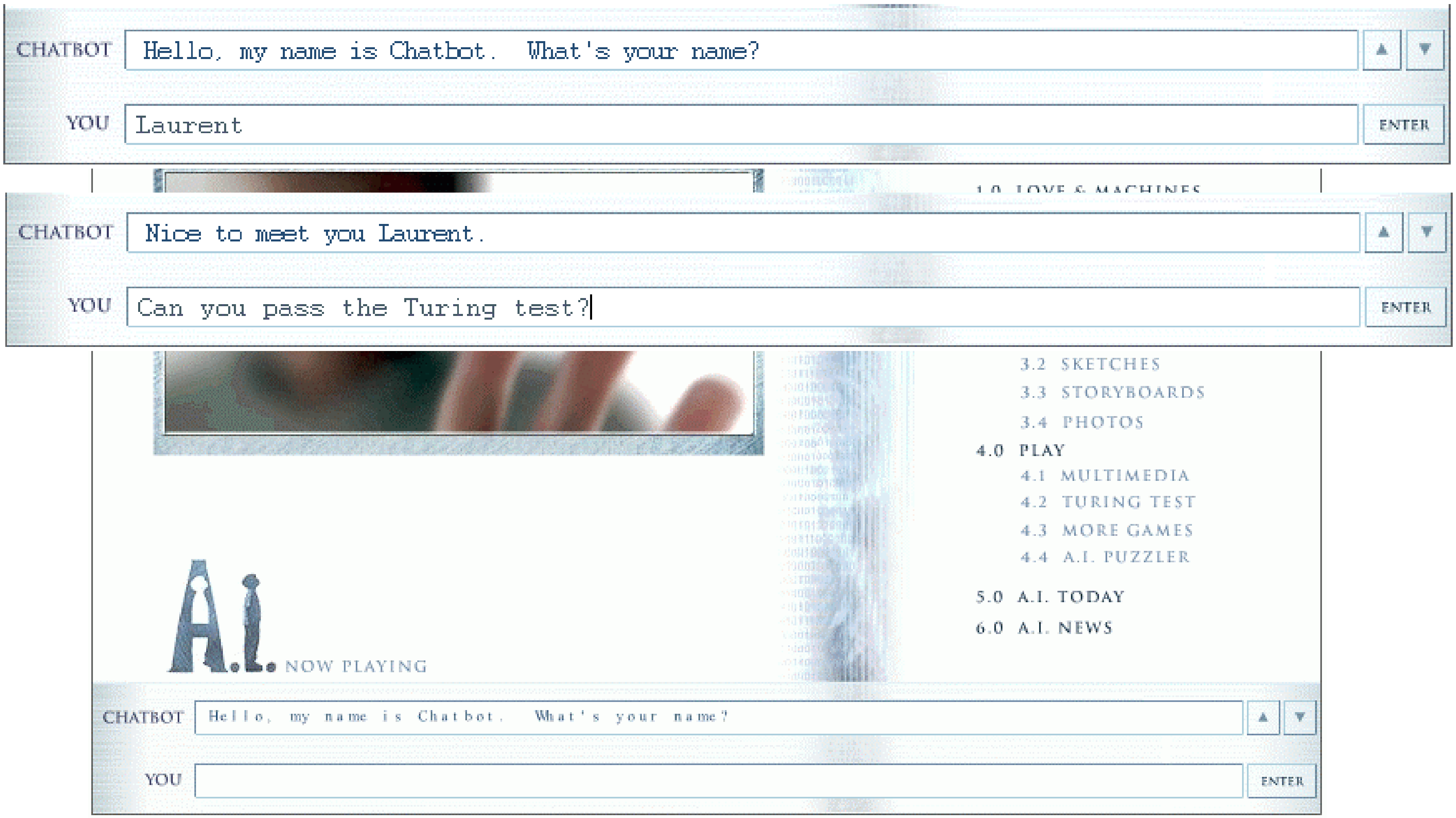
http://aimovie.warnerbros.com http://www.ai.mit.edu/projects/infolab/

Last time: The Turing Test



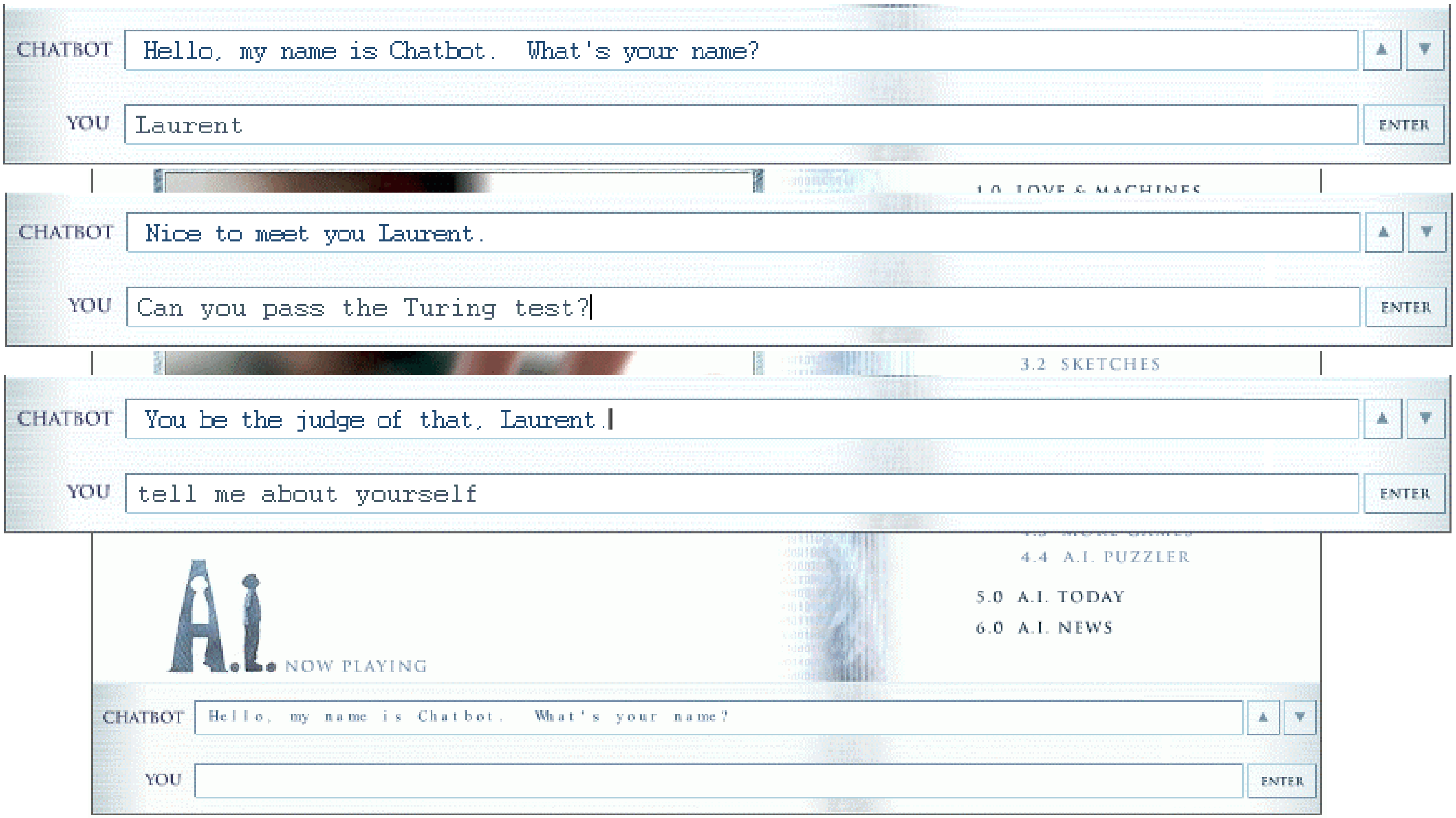
http://aimovie.warnerbros.com http://www.ai.mit.edu/projects/infolab/

Last time: The Turing Test



http://aimovie.warnerbros.com http://www.ai.mit.edu/projects/infolab/

Last time: The Turing Test



http://aimovie.warnerbros.com http://www.ai.mit.edu/projects/infolab/

Last time: The Turing Test

http://aimovie.warnerbros.com http://www.ai.mit.edu/projects/infolab/

This time: Outline

- Intelligent Agents (IA)
- Environment types
- IA Behavior
- IA Structure
- IA Types

What is an (Intelligent) Agent?

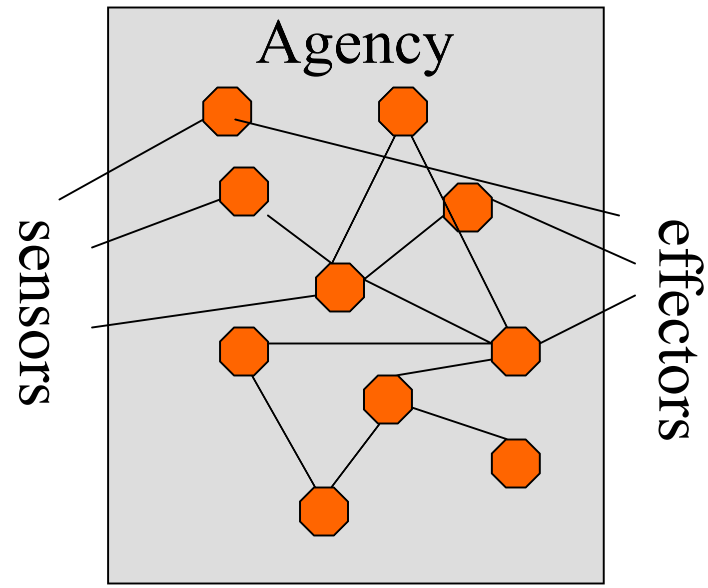
- An over-used, over-loaded, and misused term.
- Anything that can be *viewed as* **perceiving** its **environment** through **sensors** and **acting** upon that environment through its **effectors** to maximize progress towards its **goals**.
- PAGE (Percepts, Actions, Goals, Environment)
- Task-specific & specialized: well-defined goals and environment
- The notion of an agent is meant to be a tool for analyzing systems, not an absolute characterization that divides the world into agents and non-agents. Much like, e.g., object-oriented vs. imperative program design approaches.

Intelligent Agents and Artificial Intelligence

- Human mind as network of thousands or millions of agents all working in parallel. To produce real artificial intelligence, this school holds, we should build computer systems that also contain many agents and systems for arbitrating among the agents' competing results.

- Distributed decision-making and control

- Challenges:
  - Action selection: What next action to choose
  - Conflict resolution



Agent Types

We can split agent research into two main strands:

- Distributed Artificial Intelligence (DAI) – Multi-Agent Systems (MAS) (1980 – 1990)
- Much broader notion of "agent" (1990's – present)
    - interface, reactive, mobile, information

A Windshield Wiper Agent

How do we design a agent that can wipe the windshields when needed?

- Goals?
- Percepts ?
- Sensors?
- Effectors ?
- Actions ?
- Environment ?

A Windshield Wiper Agent (Cont'd)

- Goals: To keep windshields clean and maintain good visibility
- Percepts: Raining, Dirty
- Sensors: Camera (moist sensor)
- Effectors: Wipers (left, right, back)
- Actions: Off, Slow, Medium, Fast
- Environment: US inner city, freeways, highways, weather …

Towards Autonomous Vehicles

http://iLab.usc.edu

http://beobots.org

Interacting Agents

Collision Avoidance Agent (CAA)

- Goals: Avoid running into obstacles
- Percepts ?
- Sensors?
- Effectors ?
- Actions ?
- Environment: Freeway

Lane Keeping Agent (LKA)

- Goals: Stay in current lane
- Percepts ?
- Sensors?
- Effectors ?
- Actions ?
- Environment: Freeway

Interacting Agents

Collision Avoidance Agent (CAA)

- Goals: Avoid running into obstacles
- Percepts: Obstacle distance, velocity, trajectory
- Sensors: Vision, proximity sensing
- Effectors: Steering Wheel, Accelerator, Brakes, Horn, Headlights
- Actions: Steer, speed up, brake, blow horn, signal (headlights)
- Environment: Freeway

Lane Keeping Agent (LKA)

- Goals: Stay in current lane
- Percepts: Lane center, lane boundaries
- Sensors: Vision
- Effectors: Steering Wheel, Accelerator, Brakes
- Actions: Steer, speed up, brake
- Environment: Freeway

Conflict Resolution by Action Selection Agents

- **Override:** CAA overrides LKA
- **Arbitrate:** if Obstacle is Close then CAA else LKA
- **Compromise:** Choose action that satisfies both agents
- Any combination of the above
- **Challenges:** Doing the right thing

The Right Thing = The Rational Action

- **Rational Action:** The action that maximizes the expected value of the performance measure <u>given the percept sequence to date</u>

  - Rational = Best ?
  - Rational = Optimal ?
  - Rational = Omniscience ?
  - Rational = Clairvoyant ?
  - Rational = Successful ?

The Right Thing = The Rational Action

- **Rational Action:** The action that maximizes the expected value of the performance measure given the percept sequence to date

  - Rational = Best — Yes, to the best of its knowledge
  - Rational = Optimal — Yes, to the best of its abilities (incl. its constraints)
  - Rational ≠ Omniscience
  - Rational ≠ Clairvoyant
  - Rational ≠ Successful

Behavior and performance of IAs

- **Perception** (sequence) to **Action Mapping:** $f: \mathcal{P}^* \rightarrow \mathcal{A}$
  - **Ideal mapping:** specifies which actions an agent ought to take at any point in time
  - **Description:** Look-Up-Table vs. Closed Form

- **Performance measure:** a *subjective* measure to characterize how successful an agent is (e.g., speed, power usage, accuracy, money, etc.)

- (degree of) **Autonomy:** to what extent is the agent able to make decisions and actions on its own?

How is an Agent different from other software?

- Agents are **autonomous**, that is they act on behalf of the user
- Agents contain some level of **intelligence**, from fixed rules to learning engines that allow them to adapt to changes in the environment
- Agents don't only act **reactively**, but sometimes also **proactively**
- Agents have **social ability**, that is they communicate with the user, the system, and other agents as required
- Agents may also **cooperate** with other agents to carry out more complex tasks than they themselves can handle
- Agents may **migrate** from one system to another to access remote resources or even to meet other agents

Environment Types

- Characteristics
  - Accessible vs. inaccessible
  - Deterministic vs. nondeterministic
  - Episodic vs. nonepisodic
  - Hostile vs. friendly
  - Static vs. dynamic
  - Discrete vs. continuous

Environment types

Environment	Accessible	Deterministic	Episodic	Static	Discrete
Operating System					
Virtual Reality					
Office Environment					
Mars					

Environment types

Environment	Accessible	Deterministic	Episodic	Static	Discrete
Operating System	Yes	Yes	No	No	Yes
Virtual Reality	Yes	Yes	Yes/No	No	Yes/No
Office Environment	No	No	No	No	No
Mars	No	Semi	No	Semi	No

The environment types largely determine the agent design.

Structure of Intelligent Agents

- Agent = architecture + program

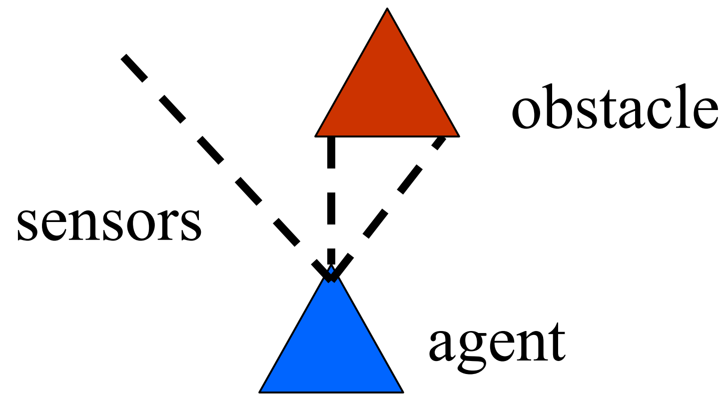
- **Agent program:** the implementation of $f: \mathcal{P}^* \rightarrow \mathcal{A}$, the agent's perception-action mapping

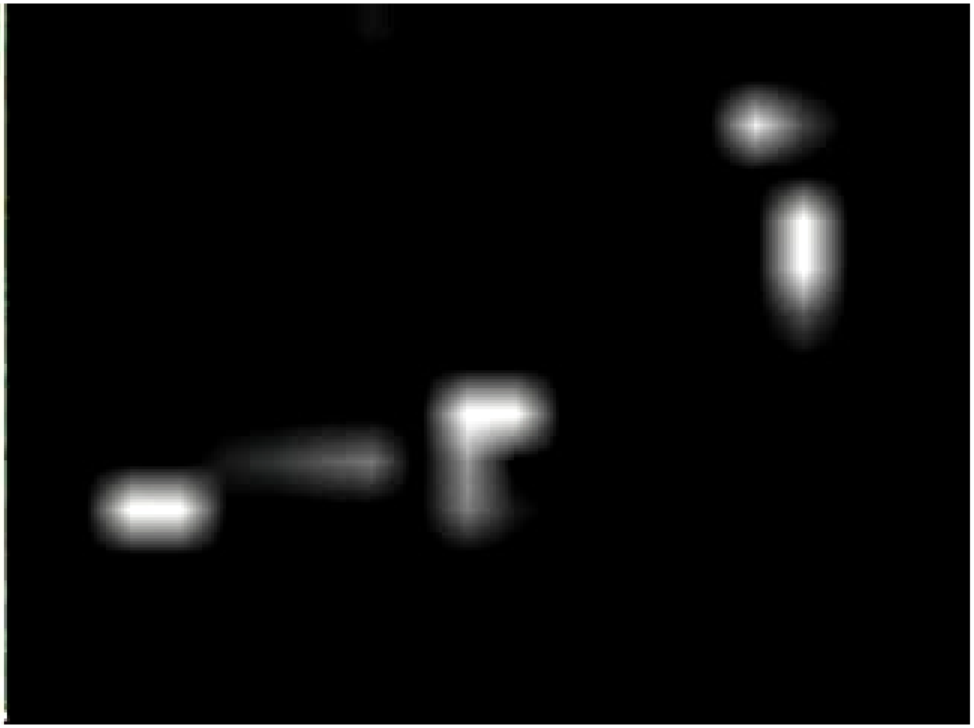
function: Skeleton-Agent(*Percept*) **returns** *Action*
memory ← UpdateMemory(memory, *Percept*)
Action ← ChooseBestAction(memory)
memory ← UpdateMemory(memory, *Action*)
return *Action*

- **Architecture:** a device that can execute the agent program (e.g., general-purpose computer, specialized device, beobot, etc.)

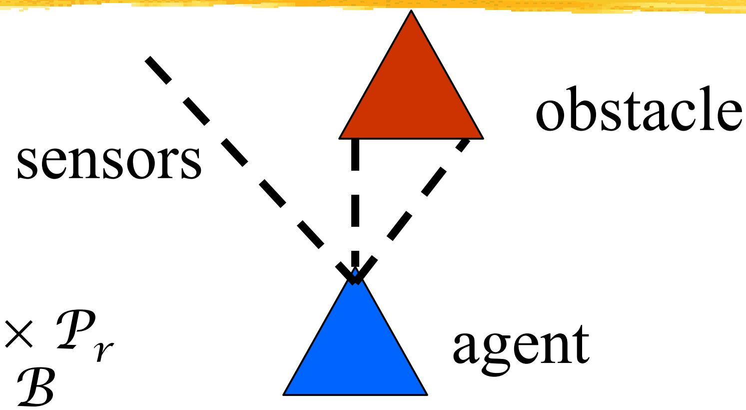
Using a look-up-table to encode $f : \mathcal{P}^* \rightarrow \mathcal{A}$

- **Example:** Collision Avoidance
  - Sensors: 3 proximity sensors
  - Effectors: Steering Wheel, Brakes
- How to generate?
- How large?
- How to select action?





Using a look-up-table to encode *f* : $\mathcal{P}^* \rightarrow \mathcal{A}$

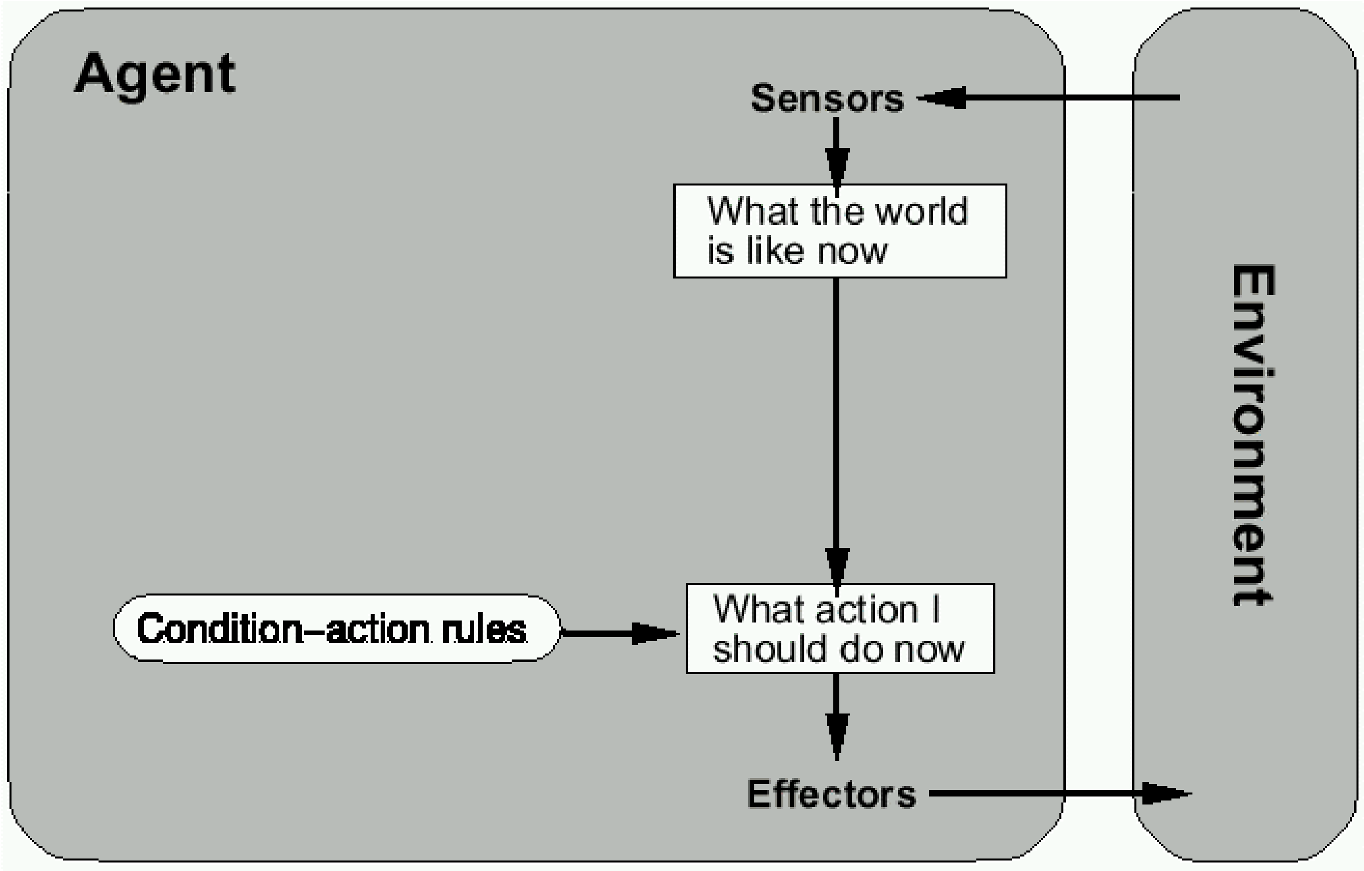


- Example: Collision Avoidance
  - Sensors: 3 proximity sensors
  - Effectors: Steering Wheel, Brakes
- How to generate: for each $p \in \mathcal{P}_\ell \times \mathcal{P}_m \times \mathcal{P}_r$ generate an appropriate action, $a \in \mathcal{S} \times \mathcal{B}$
- How large: size of table = #possible percepts times # possible actions = $|\mathcal{P}_\ell|\ |\mathcal{P}_m|\ |\mathcal{P}_r|\ |\mathcal{S}|\ |\mathcal{B}|$
  E.g., P = {close, medium, far}^{3}
  A = {left, straight, right} $\times$ {on, off}
  then size of table = 27*3*2 = 162
- How to select action? Search.

Agent types

- Reflex agents
- Reflex agents with internal states
- Goal-based agents
- Utility-based agents

Reflex agents

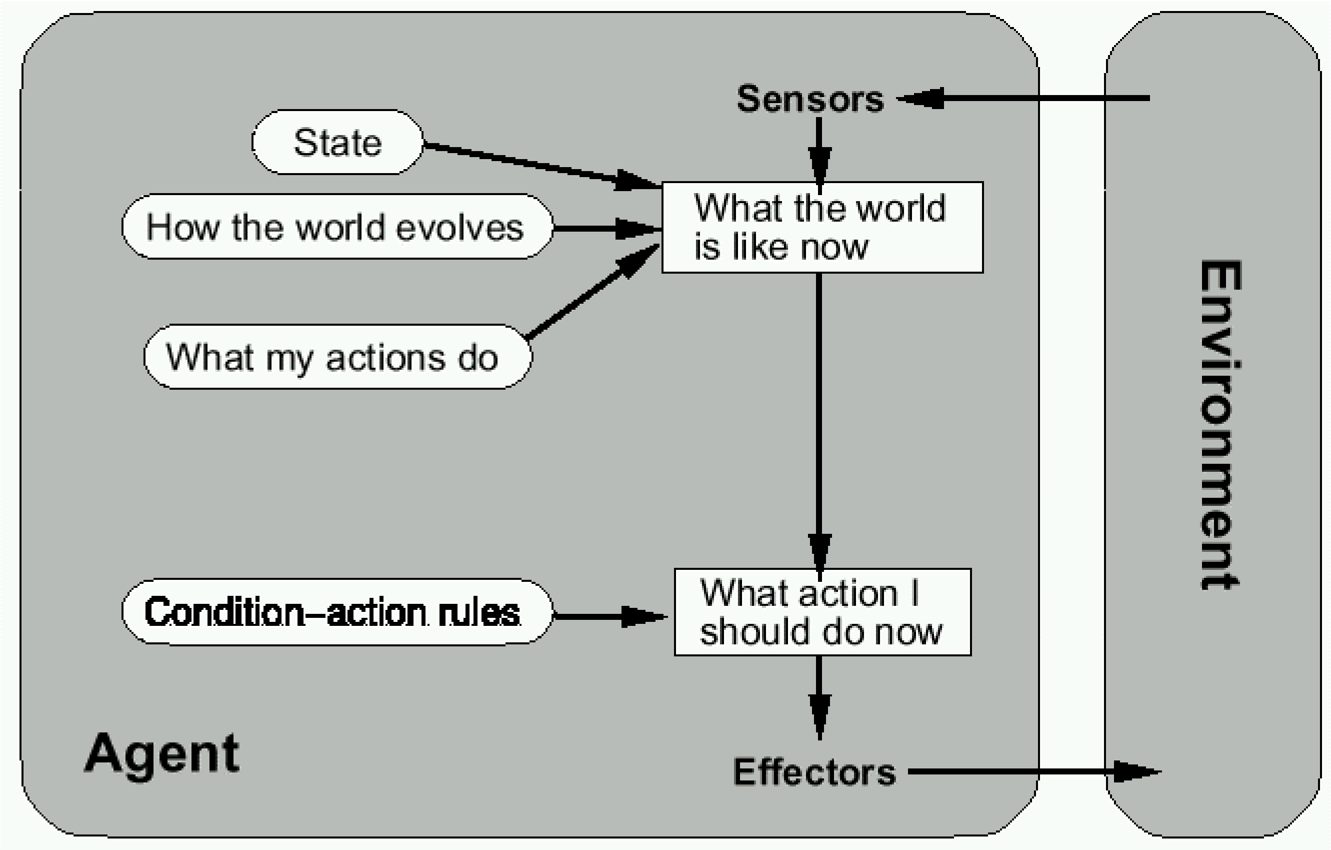


Reactive agents

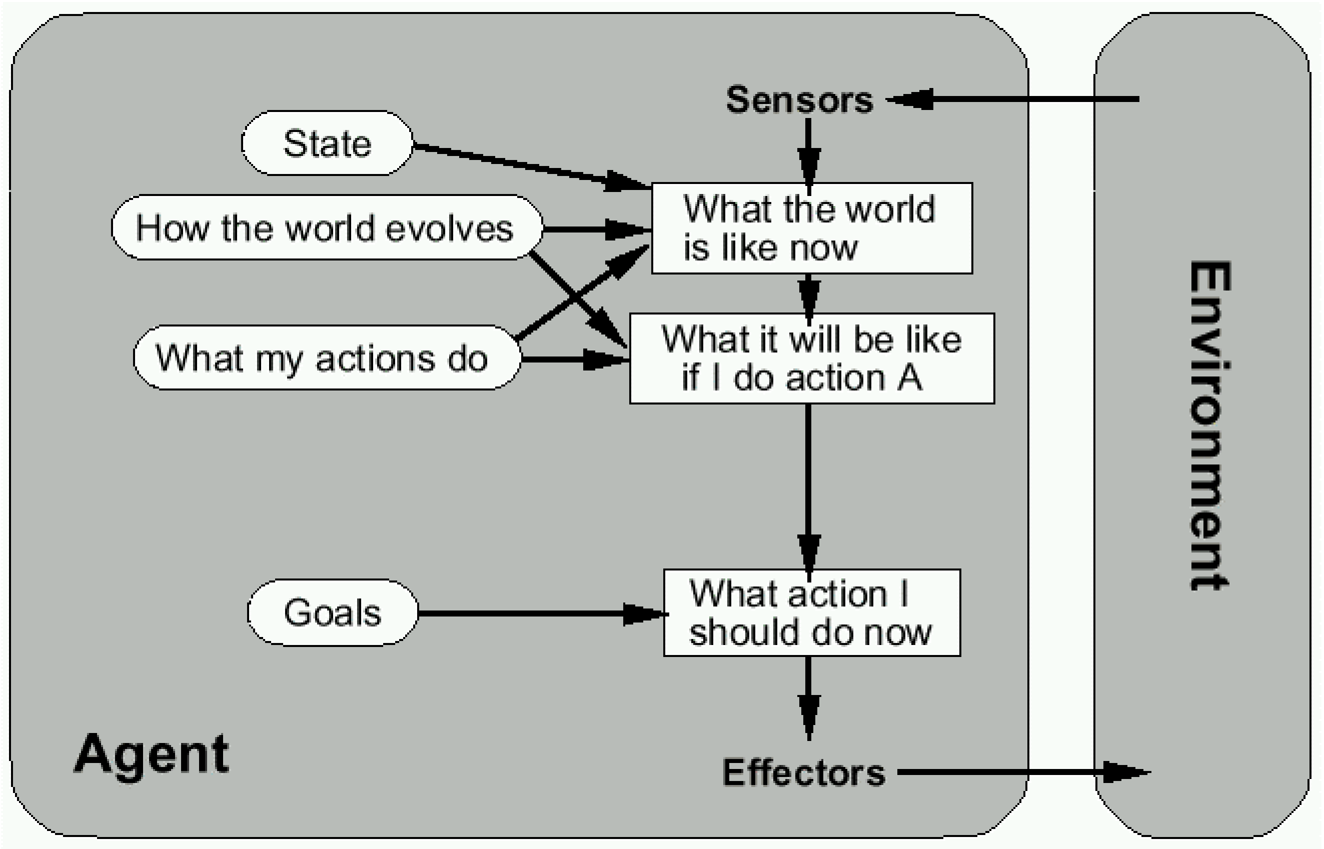
- Reactive agents do not have internal symbolic models.
- Act by stimulus-response to the current state of the environment.
- Each reactive agent is simple and interacts with others in a basic way.
- Complex patterns of behavior emerge from their interaction.

- **Benefits:** robustness, fast response time
- **Challenges:** scalability, how intelligent?
  and how do you debug them?

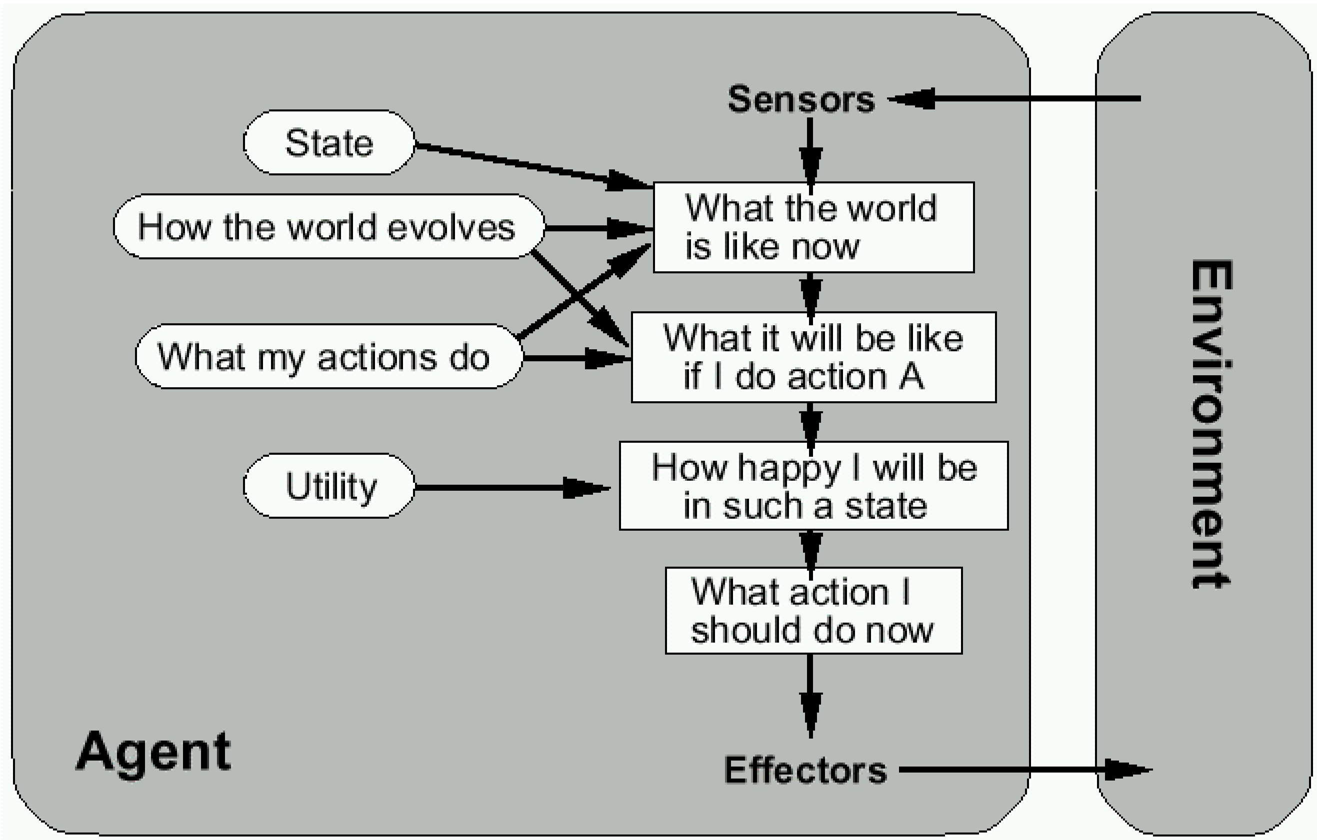
Reflex agents w/ state



Goal-based agents



Utility-based agents

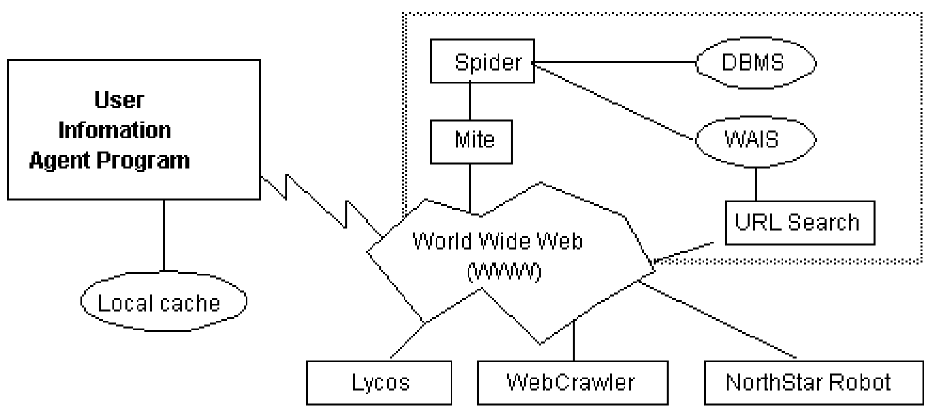


Mobile agents

- Programs that can migrate from one machine to another.
- Execute in a platform-independent execution environment.
- Require agent execution environment (places).
- Mobility not necessary or sufficient condition for agenthood.
- Practical but non-functional advantages:
  - Reduced communication cost (eg, from PDA)
  - Asynchronous computing (when you are not connected)
- Two types:
  - One-hop mobile agents (migrate to one other place)
  - Multi-hop mobile agents (roam the network from place to place)
- Applications:
  - Distributed information retrieval.
  - Telecommunication network routing.

Information agents

- Manage the explosive growth of information.
- Manipulate or collate information from many distributed sources.
- Information agents can be mobile or static.
- Examples:
  - BargainFinder comparison shops among Internet stores for CDs
  - FIDO the Shopping Doggie (out of service)
  - Internet Softbot infers which internet facilities (finger, ftp, gopher) to use and when from high-level search requests.
- Challenge: ontologies for annotating Web pages (eg, SHOE).



Summary

- **Intelligent Agents:**
  - Anything that can be *viewed as* **perceiving** its **environment** through **sensors** and **acting** upon that environment through its **effectors** to maximize progress towards its **goals**.
  - PAGE (Percepts, Actions, Goals, Environment)
  - Described as a Perception (sequence) to Action Mapping: $f: \mathcal{P}^* \rightarrow \mathcal{A}$
  - Using look-up-table, closed form, etc.

- **Agent Types:** Reflex, state-based, goal-based, utility-based

- **Rational Action:** The action that maximizes the expected value of the performance measure <u>given the percept sequence to date</u>