#TDDD92 AI-projekt

Fredrik Heintz, IDA
fredrik.heintz@liu.se
@FredrikHeintz

LINKÖPINGS UNIVERSITET

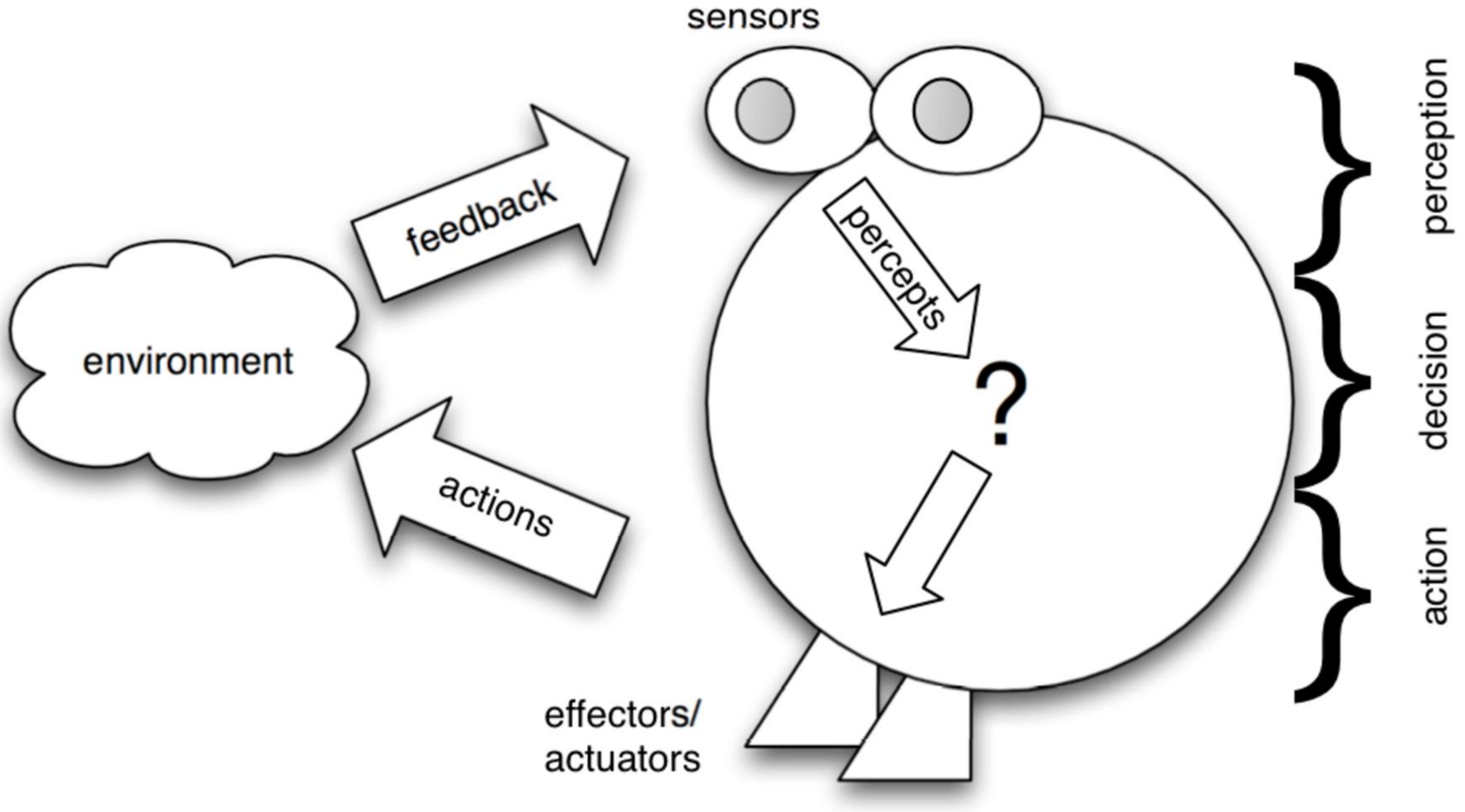
Tema: Beslutsfattande

- Ni kommer jobba med ett av de svåraste problemen inom AI: Att få robotar att fatta bra beslut.
- Att fatta beslut är svårt, särskilt om
  - man har ofullständig information
  - informationen man har är osäker och tvetydig
  - det finns andra agenter som också agerar samtidigt
  - man har begränsat med tid att fatta beslut
  - man ska fatta en sekvens av relaterade beslut

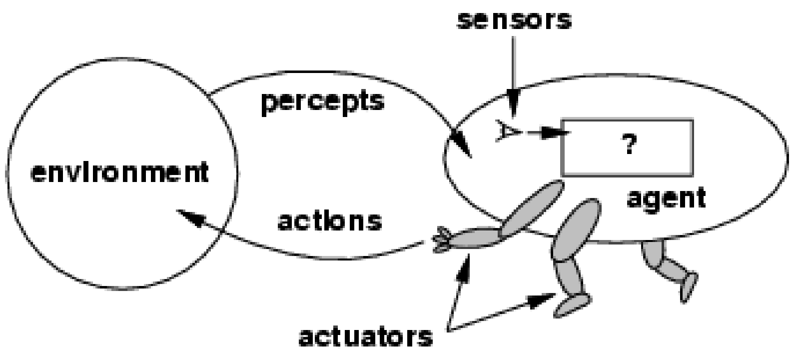
Agents

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.
- Human agent: eyes, ears, and other organs for sensors; hands, legs, mouth, and other body parts for actuators.
- Robotic agent: cameras and infrared range finders for sensors; various motors for actuators.

Decision Making



Agents and environments

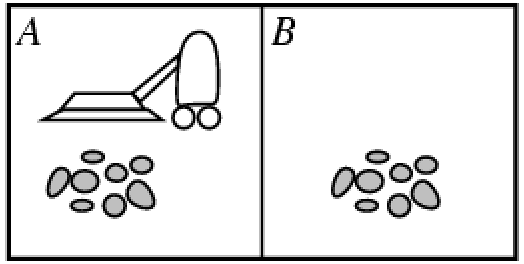


- The agent function maps from percept histories to actions:

$$[f: \mathcal{P}^{\star} \rightarrow \mathcal{A}]$$

- The agent program runs on the physical architecture to produce *f*
- agent = architecture + program

Vacuum-cleaner world



- Percepts: location and contents, e.g., [A,Dirty]
- Actions: *Left*, *Right*, *Suck*, *NoOp*

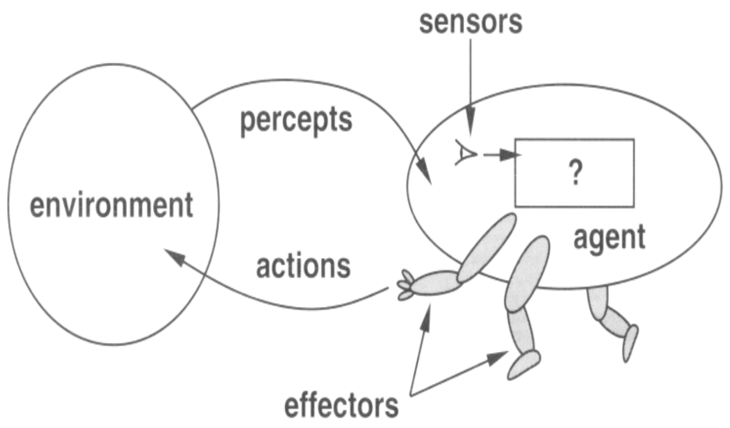
High level software control loop

1. While true {
2. **Sense** the world– (a) sensors, (b) communication, (c) supervisor input
    1. Form perceptions– (a) concept triggering, (b) propioception
    2. update beliefs (belief revision)
    3. update internal world model– (a) map, (b) localization, (c) relationships and attributes
3. **Think** about options, desires, intentions, and actions
    1. Revise desirable options and select one
    2. Deliberate about what intention to achieve next;
    3. Revise and update plan
    4. use means-ends reasoning to get a plan for the intention;
- 4. **Act**
    1. Revise intentions and select an intention to manifest
    2. execute the plan
    3. Suppress less important behaviors
    4. Start control of actuators
- 5. **Pause**
    1. until the world changes
    2. Communicate
    3. Generate and deliver user feedback

    }

The-
Frame problem
Action selection problem

Replanning problem
Envisionment problem



Rational agents

- An agent should strive to "do the right thing", based on what it can perceive and the actions it can perform. The right action is the one that will cause the agent to be most successful.
- Performance measure: An objective criterion for success of an agent's behavior.
- E.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

Rational agents

- Rational Agent: For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.
- Rationality is distinct from omniscience (all-knowing with infinite knowledge).
- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration).
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt).

PEAS – Performance measure, Environment, Actuators, Sensors

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design
- Consider, e.g., the task of designing an automated taxi driver:
  - Performance measure: Safe, fast, legal, comfortable trip, maximize profits
  - Environment: Roads, other traffic, pedestrians, customers
  - Actuators: Steering wheel, accelerator, brake, signal, horn
  - Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

Environment types

- Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.
- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)
- Episodic (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

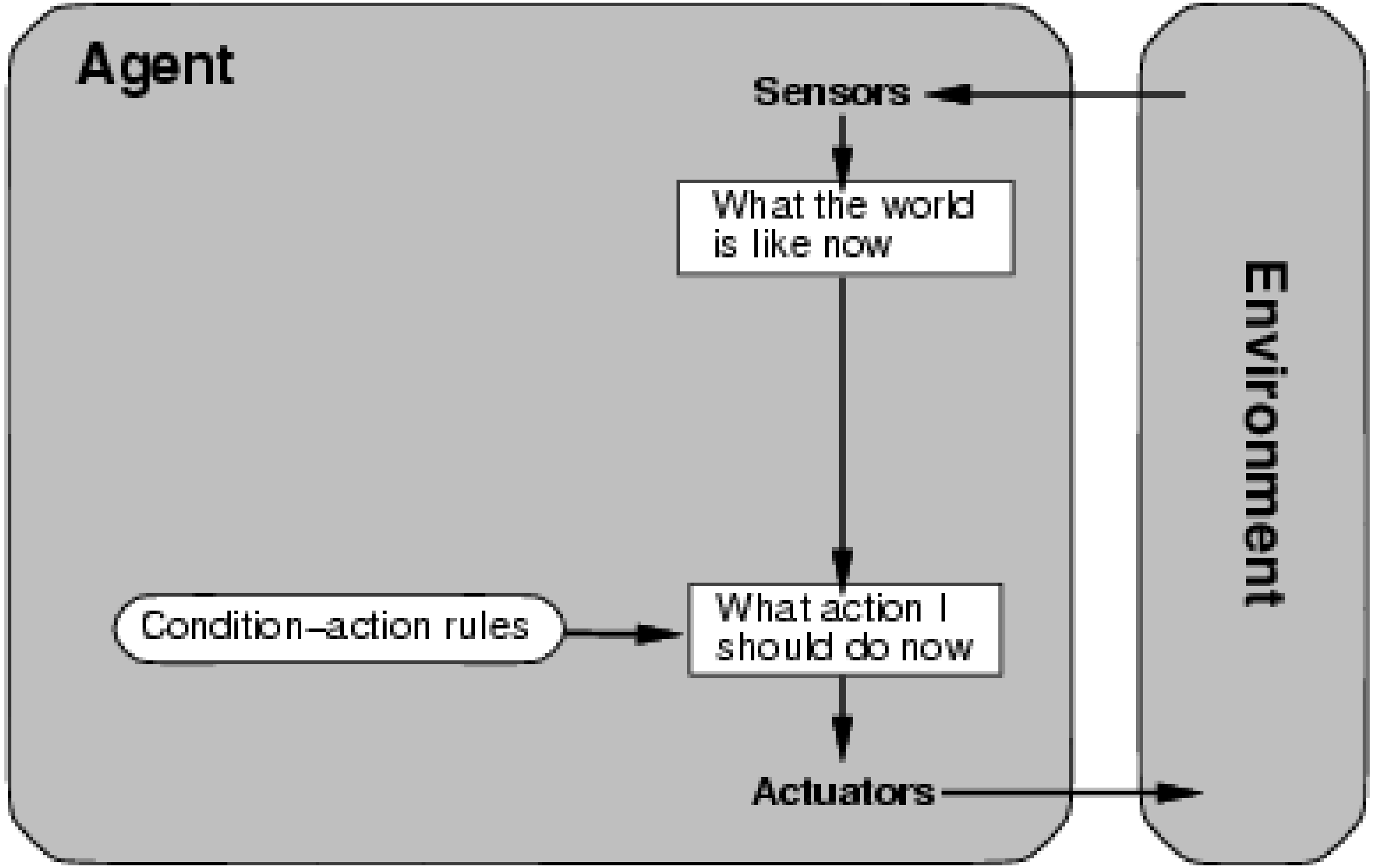
Environment types

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating. (The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does)
- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions.
- Single agent (vs. multiagent): An agent operating by itself in an environment.

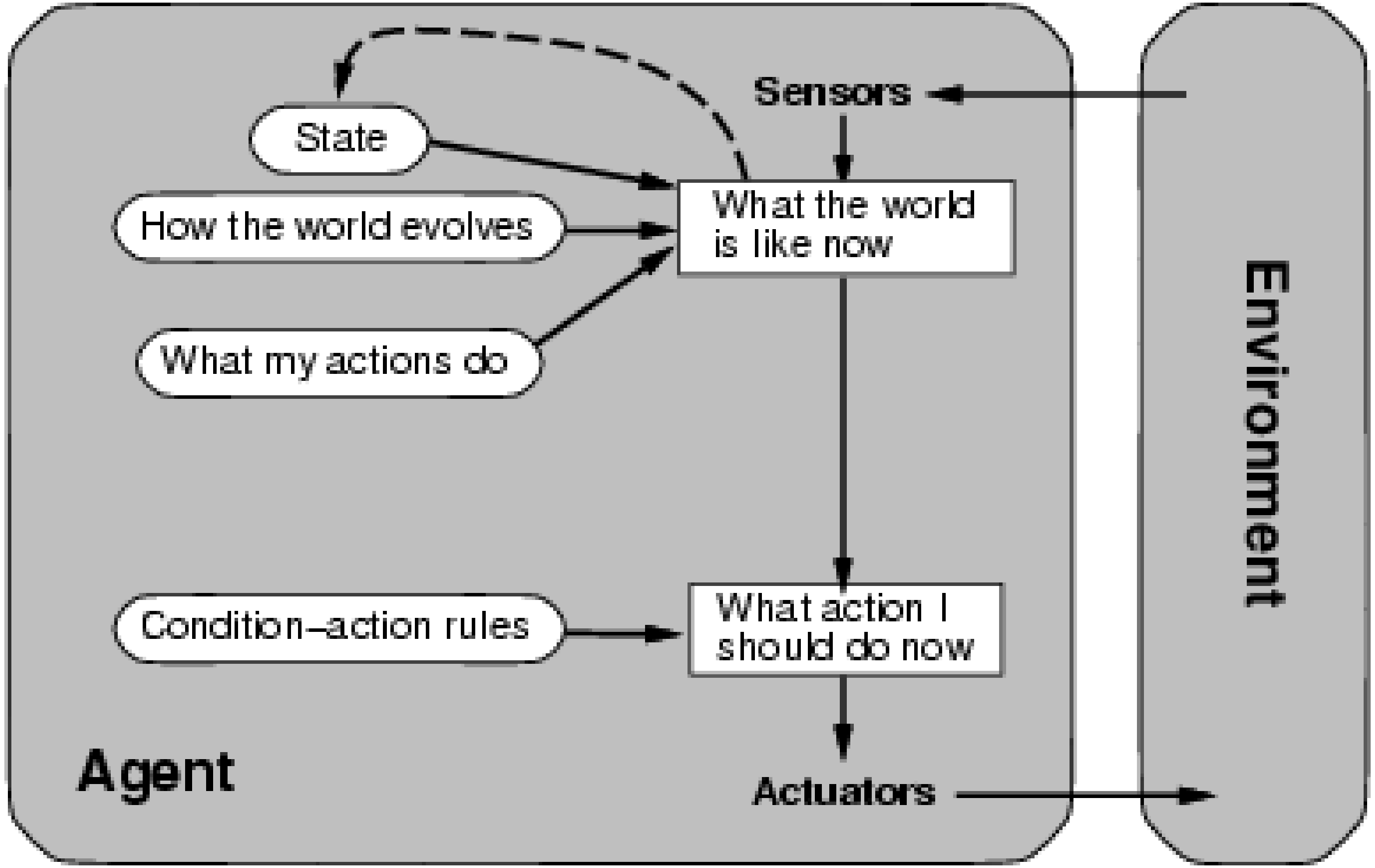
Agent types

- Four basic types in order of increasing generality:
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents

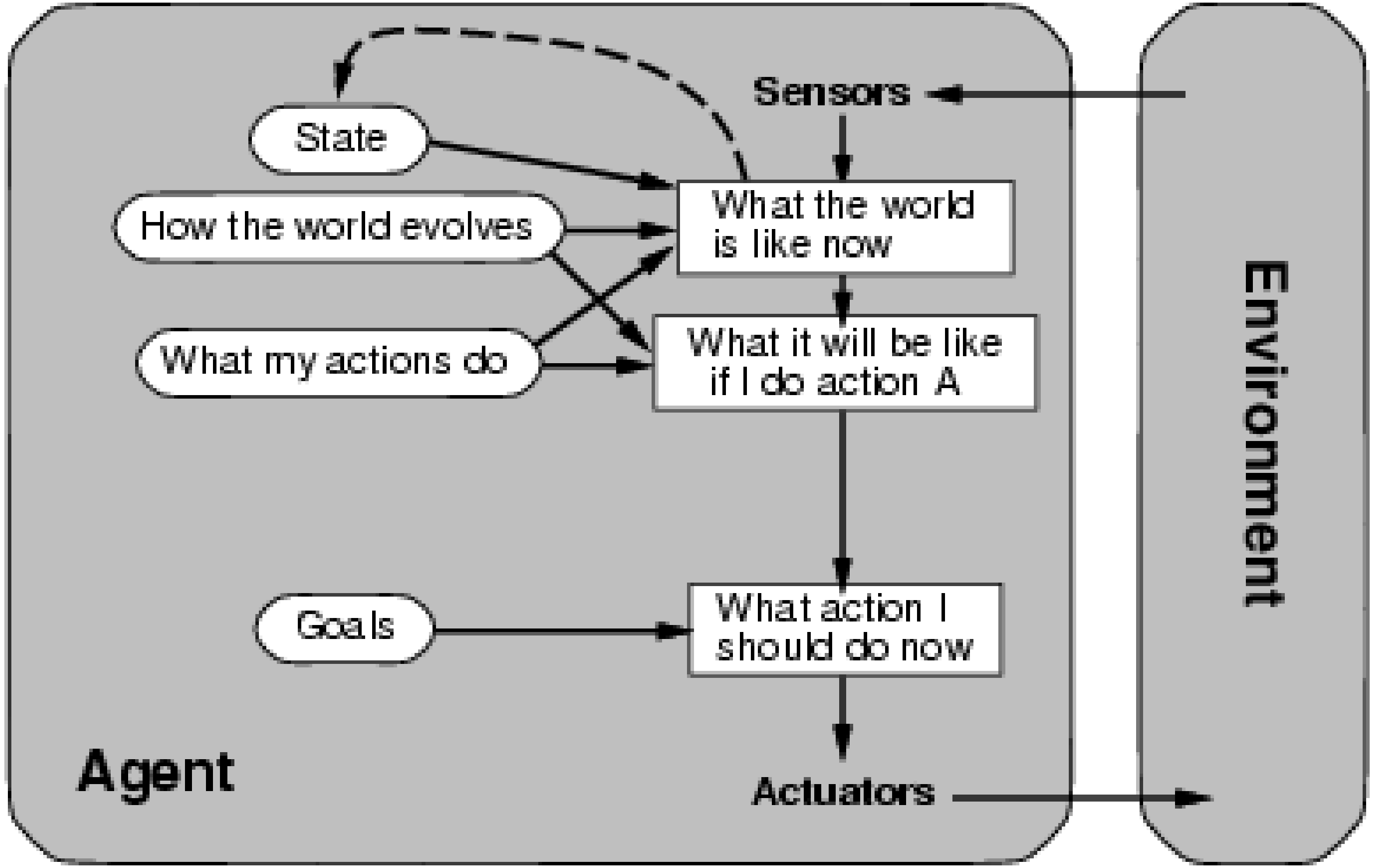
Simple reflex agents



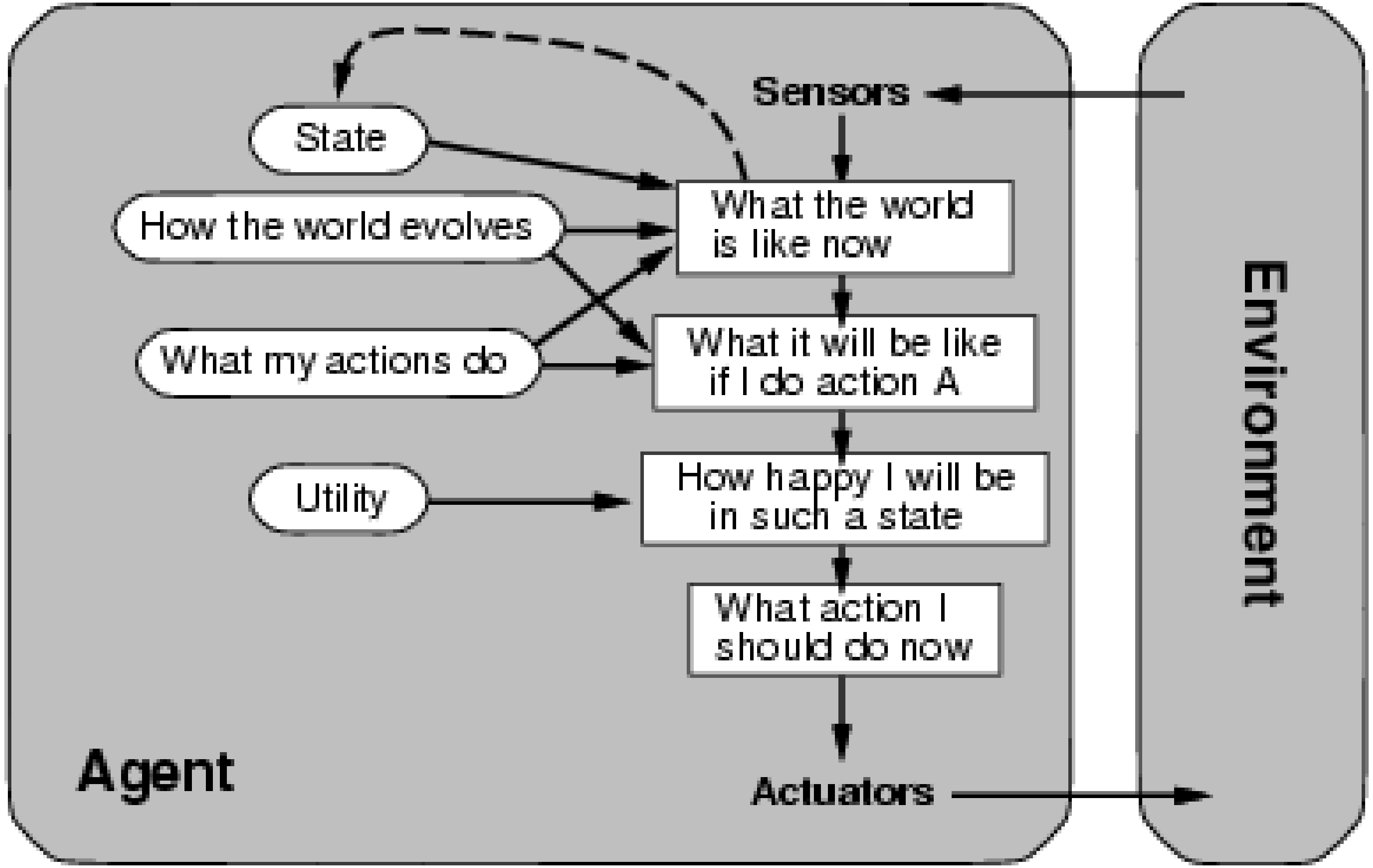
Model-based reflex agents



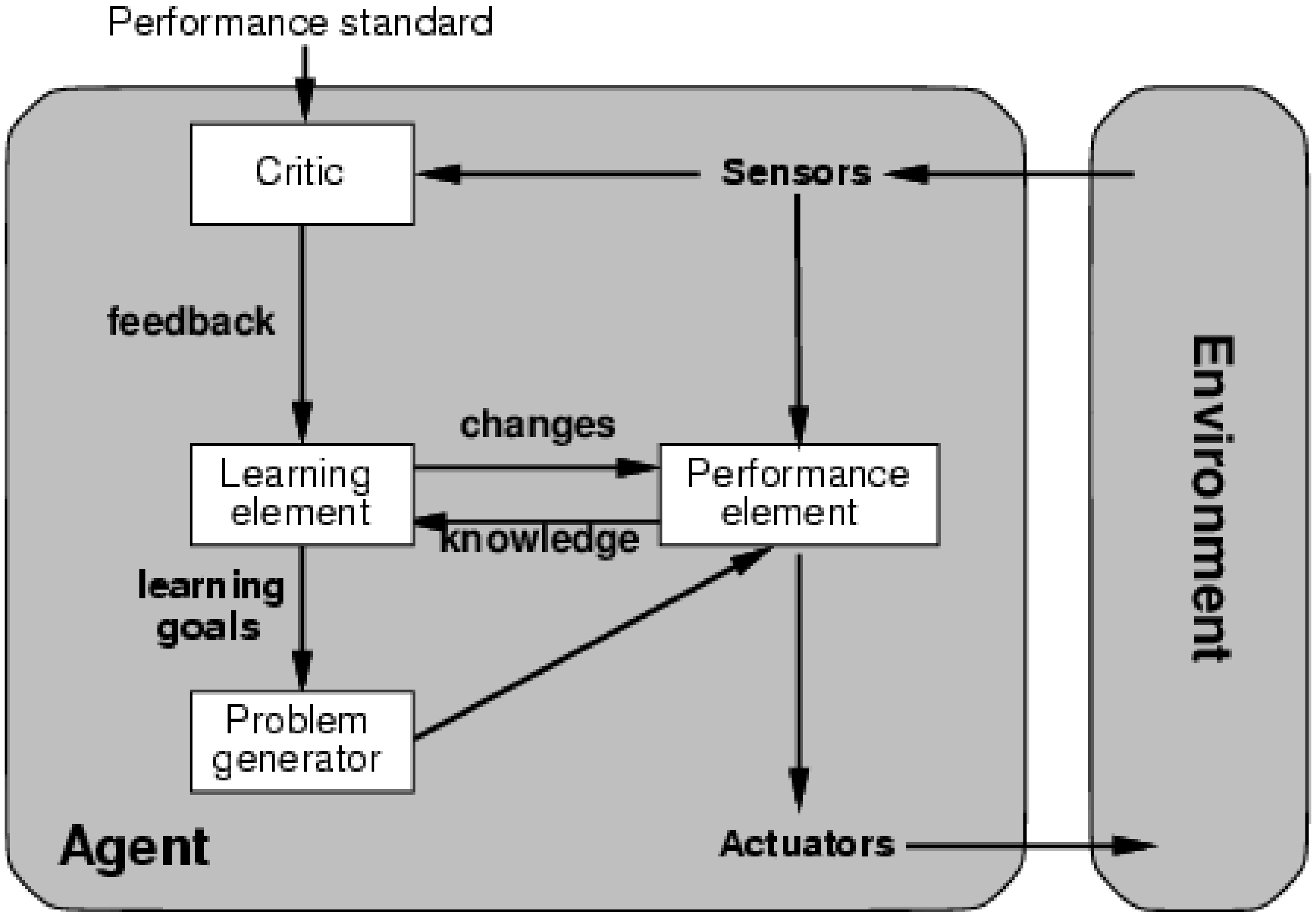
Goal-based agents



Utility-based agents



Learning agents



Individual Decision Making

- Explicit decision making
  - Decision trees
  - Rules
  - Automata
  - Single agent task specification languages
- Decision theoretic decision making
  - Markov Decision Processes (MDP)
  - Partially Observable Markov Decision Processes (POMDP)
- Declarative decision making
  - Planning
  - Theorem Proving, Constraint satisfaction

Multi-agent Decision Making

- Explicit
  - Mutual modeling
  - Norms
  - Organizations and Roles
  - Multi-agent task specification languages
- Decision theoretic: Decentralized POMDPs (Dec-POMDP)
- Game theoretic: Auctions
- Declarative
  - Multi-agent planning
  - Distributed constraint satisfaction

Practical Reasoning and Belief-Desire-Intention Agents

Practical Reasoning

"Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes."

Bratman

Practical Reasoning

- Human practical reasoning consists of two activities:
  - *deliberation* - deciding what state of affairs we want to achieve;
  - *means-ends reasoning* - deciding how to achieve these states of affairs.
- The output of deliberation is *intentions*.

Means-Ends Reasoning

Given

- representation of goal/intention to achieve;
- representation of actions it can perform; and
- representation of the environment generate a plan to achieve the goal.

Essentially this is *planning*.

Implementing Practical Reasoning Agents

```
1. while true
2.   observe the world
3.   update internal world model
4.   deliberate about what
       intentions to achieve next
5.   use means-ends reasoning to
       get a plan for the intention
6.   execute the plan
7. end while
```

Agent Control Loop Version 1

```
B := B0;
while true do
  get next percept p;
  B := brf(B,p);
  I := deliberate(B);
  P := plan(B,I);
  execute(P);
end while
```

Deliberation

- How does an agent deliberate?
  - begin by trying to understand what the options available to you are;
  - choose between them, and commit to some.

Chosen options are then intentions.

Deliberation cont.

- The deliberate function can be decomposed into two distinct functional components:
  - *option generation*
    - in which the agent generates a set of possible alternatives; and
    - represent option generation via a function, options, which takes the agent's current beliefs and current intentions, and from them determines a set of options (= desires).
  - *filtering*
    - in which the agent chooses between competing alternatives, and commits to achieving them.
    - In order to select between competing options, an agent uses a filter function.

Agent Control Loop Version 2

```
B := B_0;
I := I_0;
while true do
  get next percept p;
  B := brf(B,p);
  D := options(B,I);
  I := filter(B,D,I);
  P := plan(B,I);
  execute(P);
end while
```

Commitment Strategies

- *Blind commitment* - A blindly committed agent will continue to maintain an intention until it believes the intention has actually been achieved. Blind commitment is also sometimes referred to as fanatical commitment.
- *Single-minded commitment* - A single-minded agent will continue to maintain an intention until it believes that either the intention has been achieved, or else that it is no longer possible to achieve the intention.
- *Open-minded commitment* - An open-minded agent will maintain an intention as long as it is still believed possible.

Commitment

- An agent has commitment both to *ends* (i.e. of wishes to bring about), and *means* (i.e., the mechanism via which the agent wishes to achieve the state of affairs).
- Currently, our agent control loop is overcommitted, both to means and ends.
  Modification: *replan* if ever a plan goes wrong.

Agent Control Loop Version 3

```
B := B_0; I := I_0;
while true do
  get next percept p;
  B := brf(B,p); D := options(B,I);
  I := filter(B,D,I); P := plan(B,I);
  while not empty(P) do
    a := first(P); execute(a); P := rest(P);
    get next percept p; B := brf(B,p);
    if not sound(P,B,I) then P := plan(B,I);
  end while
end while
```

Commitment

- Still overcommitted to intentions: never stops to consider whether or not its intentions are appropriate.
- Modification: stop to determine whether intentions have succeeded or whether they are impossible (single-minded commitment).

Agent Control Loop Version 4

```
B := B_0; I := I_0;
while true do
  get next percept p;
  B := brf(B,p); D := options(B,I);
  I := filter(B,D,I); P := plan(B,I);
  while not ( empty(P) or succeeded(B,I) or
              impossible(B,I) ) do
    a := first(P); execute(a); P := rest(P);
    get next percept p; B := brf(B,p);
    if not sound(P,B,I) then P := plan(B,I);
  end while
end while
```

Intention Reconsideration

- Our agent gets to reconsider its intentions once every time around the outer control loop, i.e., when:
  - it has completely executed a plan to achieve its current intentions; or
  - it believes it has achieved its current intentions; or
  - it believes its current intentions are no longer possible.
- This is limited in the way that it permits an agent to reconsider its intentions.
- Modification: Reconsider intentions after executing every action.

Agent Control Loop Version 5

```
B := B0; I := I0;
while true do
 get next percept p;
 B := brf(B,p); D := options(B,I);
 I := filter(B,D,I); P := plan(B,I);
 while not ( empty(P) or succeeded(B,I) or
             impossible(B,I) ) do
  a := first(P); execute(a); P := rest(P);
  get next percept p; B := brf(B,p);
  D := options(B,I); I := filter(B,D,I);
  if not sound(P,B,I) then P := plan(B,I);
 end while
end while
```

Intention Reconsideration

- But intention reconsideration is *costly*! A dilemma:
  - an agent that does not stop to reconsider its intentions sufficiently often will continue attempting to achieve its intentions even after it is clear that they cannot be achieved, or that there is no longer any reason for achieving them;
  - an agent that *constantly* reconsiders its attentions may spend insufficient time actually working to achieve them, and hence runs the risk of never actually achieving them.
- Solution: incorporate an explicit *meta-level control* component, that decides whether or not to reconsider.

Agent Control Loop Version 6

```
B := B_0; I := I_0;
while true do
  get next percept p;
  B := brf(B,p); D := options(B,I);
  I := filter(B,D,I); P := plan(B,I);
  while not ( empty(P) or succeeded(B,I) or
              impossible(B,I) ) do
    a := first(P); execute(a); P := rest(P);
    get next percept p; B := brf(B,p);
    if reconsider(B,I) then
      D := options(B,I); I := filter(B,D,I);
    if not sound(P,B,I) then P := plan(B,I);
  end while
end while
```

Optimal Intention Reconsideration

- Kinny and Georgeff's experimentally investigated effectiveness of intention reconsideration strategies.
- Two different types of reconsideration strategy were used:
  - *bold* agents never pause to reconsider intentions, and
  - *cautious* agents stop to reconsider after every action.
- Dynamism in the environment is represented by the *rate of world change*, γ.

Kinny and Georgeff's Results

- If γ is low (i.e., the environment does not change quickly), then bold agents do well compared to cautious ones. This is because cautious ones waste time reconsidering their commitments while bold agents are busy working towards - and achieving - their intentions.
- If γ is high (i.e., the environment changes frequently), then cautious agents tend to outperform bold agents. This is because they are able to recognize when intentions are doomed, and also to take advantage of serendipitous situations and new opportunities when they arise.

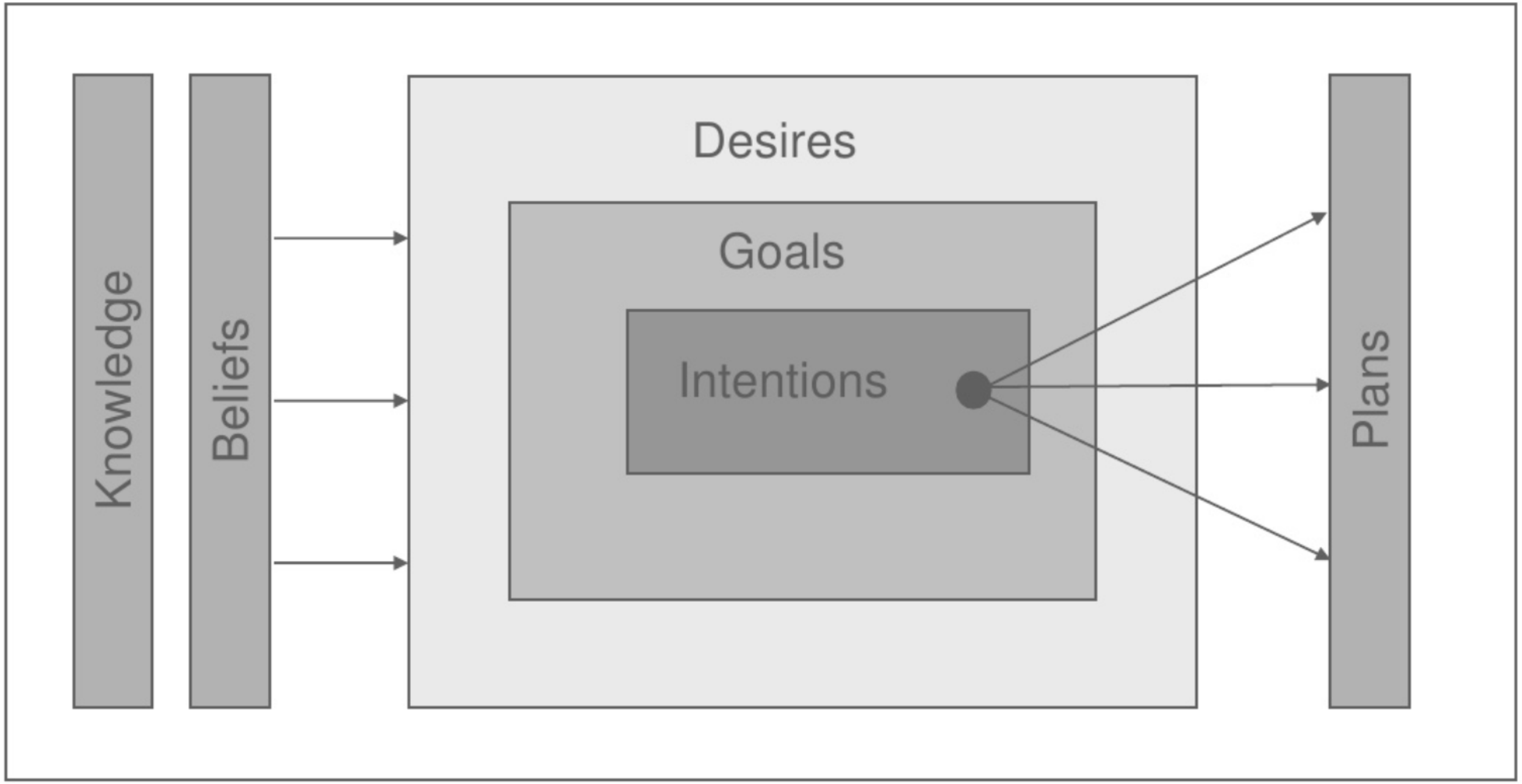
Belief Desire Intention (BDI) Architecture

- **Beliefs:**
  - Agent's view of the environment/world.

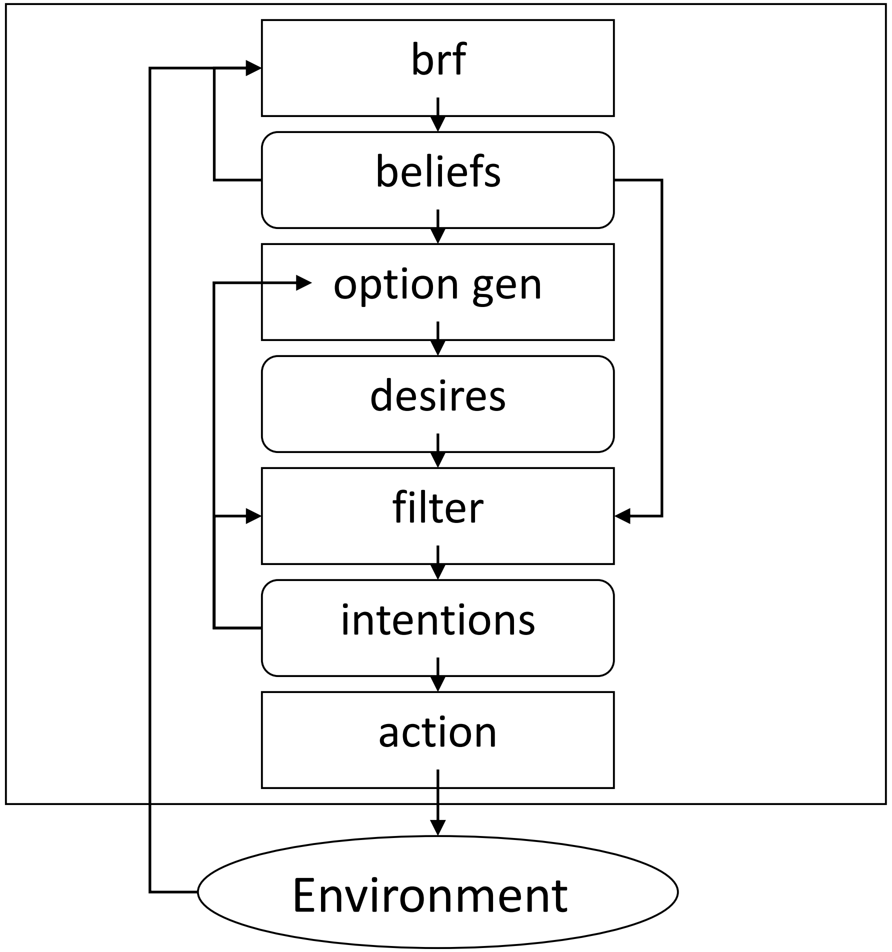
- **Desires:**
  - Follow from the beliefs. Desires can be unrealistic and inconsistent.
- **Goals:**
  - A subset of the desires. Realistic and consistent. Determine potential processing.
- **Intentions:**
  - A subset of the goals. A goal becomes an intention when an agent decides to commit to it (e.g. by assigning priorities to goals)

- **Plans:**
  - Sequences of actions that are needed to achieve the intentions, given the agent's beliefs

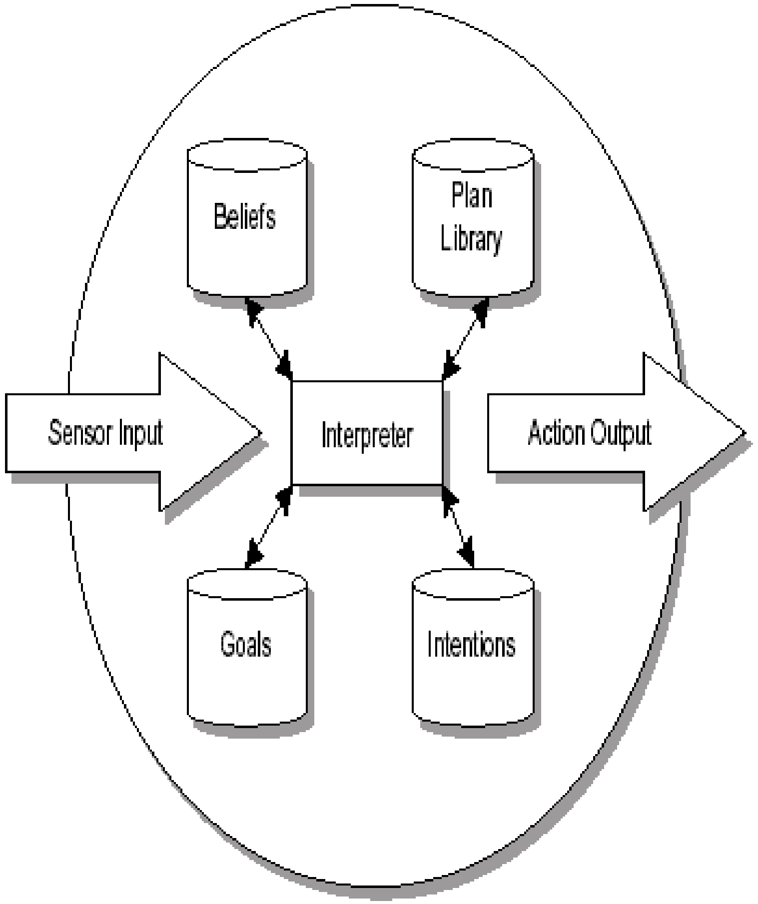
Belief Desire Intention (BDI) Architecture



Belief Desire Intention (BDI) Architecture



Procedural Reasoning System (PRS)



Reinforcement Learning

Kursupplägg

Examination

- UPG1 Projekt (U, 3, 4, 5) **deadline 7/1 2018**
  - Grupp om 4-6 personer
  - Ni definierar själva målet med ert projekt
  - Ni ska utvärdera hur bra er lösning är
  - Betyg baseras på hur svår uppgiften är och hur väl genomförd den är
- UPG2 Individuell skriftlig rapport (U, 3, 4, 5) **deadline 29/10**
  - Var och en ska göra en individuell förstudie inför projektet
  - Ska utgå från vetenskapliga artiklar
  - Ska ha en utredande karaktär, t.ex. sammanfatta eller utvärdera

Kursupplägg

- En (delvis) bemannad labb varje vecka, mån 13-17.
  - I humanoidlabbet
  - Assistenter är Mattias Tiger och Fredrik Löfgren
- Ett projektmöte/seminarie varje vecka, fre 13-15.
  - Kort statusrapport från varje grupp, både tekniskt och introspektivt
  - Huvudsakligen för er skull!
  - Ni får gärna be mig presentera/diskutera specifika saker
  - **Till nästa vecka:**
    dela in er i grupper och fundera på vad ni vill göra

UPG1 Projektarbete

- *Skriva en kort projektbeskrivning, c:a 1 A4-sida som innehåller följande:*
  - Mål ska beskriva vad ni önskar uppnå.
  - Syfte ska beskriva varför det valda målet är relevant.
  - Genomförande beskriver väldigt kortfattat hur ni planerar att uppnå målet, t.ex. vilka tekniker tänker ni använda.
  - Utvärderingen ska beskriva hur ni tänker utvärdera att ni har uppnått målet.
- **Deadline för att lämna in projektbeskrivningen är söndag 5/11.**
- *Presentera planerna för ert projekt* ***fredag 3/11****.*
- *Genomför projektet på ett självgående sätt. Ge statusrapport på de veckovisa seminarierna med start* ***fredag 10/11****.*
- *Presentera och demonstrera det färdiga projektet* ***måndag 18/12****.*
- *Skriv en teknisk rapport.* **Deadline söndag 7/1 2018.**
- Betyget baseras på hur svår uppgiften är, hur bra er lösning är, hur väl ni har utvärderat ert system samt hur välskriven den tekniska rapporten är.

UPG2 Individuell skriftlig rapport

- *Skriv en kort projektbeskrivning på c:a 1 A4-sida.*
  - Beskrivningen ska definiera problemet ni önskar lösa (mål),
  - anledningen till att det är viktigt att lösa det givna problemet (syfte),
  - de tekniker ni tänker utvärdera inklusive referenser till de vetenskapliga artiklar ni tänker utgå ifrån,
  - samt enligt vilka kriterier ni tänker utvärdera teknikerna.
- **Deadline för att lämna in projektbeskrivningen söndag 24/9.**
- *Presentera de individuella projektet gruppvis* ***fredag 13/10****.*
- *Skriv en utvärderingsrapport på 4-6 sidor*. **Deadline söndag 29/10.**
- Betyget baseras på hur tydligt och välmotiverat problemet är, hur tekniskt avancerade tekniker som valts, hur väl utvärderingen är gjord och hur väl rapporten är skriven.

Techniques / Problems

- Machine learning
  - Classifiers / Predictors
    - Decision trees
    - Neural networks
    - SVM
    - Bayesian networks
  - Action selection
    - Reinforcement learning
    - Dynamic Bayesian networks
  - Genetic Algorithms
- Planning and Coordination
  - Positioning / Potential fields
  - Path planning
  - Task allocation
  - Task planning
  - Monte Carlo Tree Search
  - Game Theory
- Spatio-Temporal Reasoning
  - Opponent modeling
  - Activity recognition
- Communication

Example Topics and References

- General
  - S. Ruseel and P. Norvig. Artificial Intelligence, A Modern Approach, 4th Edition.
- Reinforcement Learning
  - P. Stone and R. Sutton, Scaling reinforcement learning toward robocup soccer. In Proceedings of the International Conference on Machine Learning.
  - P. Stone and R.S. Sutton, Keepaway Soccer: A Machine Learning Testbed.
  - Martin Lauer, M. R. (2000). *An Algorithm for Distributed Reinforcement Learning in Cooperative Multi-Agent Systems*. Department of Computer Science, University of Karlsruhe.
  - T.Hester, M.Quinlan and Peter Stone, Generalized Model Learning for Reinforcement Learning on a Humanoid Robot. IEEE International Conference on Robotics and Automation (ICRA). 2010.
- Random Forests
  - Breiman, Leo (2001). *Random Forests*. Machine Learning 45 (1): 5–32.

Example Topics and References

- Planning and Coordination
  - S. Coradeschi and L. Karlsson, A Behavior-Based Approach to Reactivity and Coordination.
  - P. Stone and M. Veloso, Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork.
  - C. Erdogan and M. Veloso. Action Selection via Learning Behavior Patterns in Multi-Robot Domains. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (2011).
  - Mathijs de Weerdt, A. t. (u.d.). *Multi-agent Planning, An introduction to planning and coordination.* Dept. of Software Technology, Delft University of Technology.
  - R. Subramanian, R.Geetharian, P.Prathima and M.Sangeetha, Decision Tree Based Task Allocation Algorithm For Soccer Domain.
  - J. Kok, M. Spaan and N. Vlassis. Non-communicative multi-robot coordination in dynamic environments. Robotics and Autonomous Systems 50. 2005.
  - A. Gerevini et al. Combining Domain-Independent Planning and HTN Planning: The Duet Planner. 2008.
  - U. Kuter and D. Nau. Forward-chaining planning in nondeterministic domains". Proc. AAAI. 2004.
  - D. Nau et al. SHOP2: An HTN Planning System. Journal of Artificial Intelligence Research 20 (2003).
- Opponent modelling
  - P. Stone, P. Riley, and M. Veloso. Defining and Using Ideal Teammate and Opponent Agent Models: A Case Study in Robotic Soccer. Proceedings International Conference on MultiAgent Systems (2000).

Example Topics and References

- Task Allocation
  - D. Vail and M. Veloso. Multi-Robot Dynamic Role Assignment and Coordination Through Shared Potential Fields.
  - The Power of Sequential Single-Item Auctions for Agent Coordination. Proc AAAI.
  - Shih-An Li et al. Task Allocation Design for Autonomous Soccer Robot. Proceedings of Intelligent Robotics Systems (IROS). 2013.
  - A. Campbell and A. Wu. Multi-agent role allocation: issues, approaches, and multiple perspectives. Autonomous Agents and Multi-Agent Systems 22(2). 2011.
- Path Planning
  - S. Rodríguez et al. Fast Path Planning Algorithm for the RoboCup Small Size League. Proceedings of RoboCup Symposium (2014).
  - S. LaValle and J. Kuffner. Rapidly-Exploring Random Trees: Progress and Prospets.
  - Plaku and Karaman. Motion planning with temporal-logic specifications: Progress and challenges. AI Communications 29(1). 2016.

Example Topics and References

- Monte Carlo Tree Search
  - Cameron Browne, E. P. (2012). *A Survey of Monte Carlo Tree Search Methods*. IEEE Transactions on Computational Intelligence and AI in Games 4(1).
  - H. Baier and M. Winands. (2015). *MCTS Minimax Hybrids*. IEEE Transactions on Computational Intelligence and AI in Games. 7(2).
  - S., Spyridon, D. Robles, and S. Lucas. (2011). *Fast approximate Max-N Monte Carlo Tree Search for MS Pacman*. IEEE Transactions on Computational Intelligence and AI in Games. 3(2).
  - Levente Kocsis, C. S. (u.d.). *Improved Monte-Carlo Search*. University of Tartu.
- Game Theory
  - Game-Theoretic Control for Robot Teams - Rosemary Emery-Montemerlo, 2005
- Synthesis of Finite State Machines
  - S. Maoz and J. Ringert. GR(1) synthesis for LTL specification patterns. Proc. Foundations for Software Engineering. 2015.
  - M. Guo and D. Dimarogonas. Multi-agent plan reconfiguration under local LTL specifications. International Journal of Robotics Research 34(2). 2015.

Hela kursen är experimentell och agil
– vi testar oss fram och anpassar allt eftersom!

#TDDD92 AI-projekt

Fredrik Heintz, IDA
fredrik.heintz@liu.se
@FredrikHeintz

LINKÖPINGS
UNIVERSITET