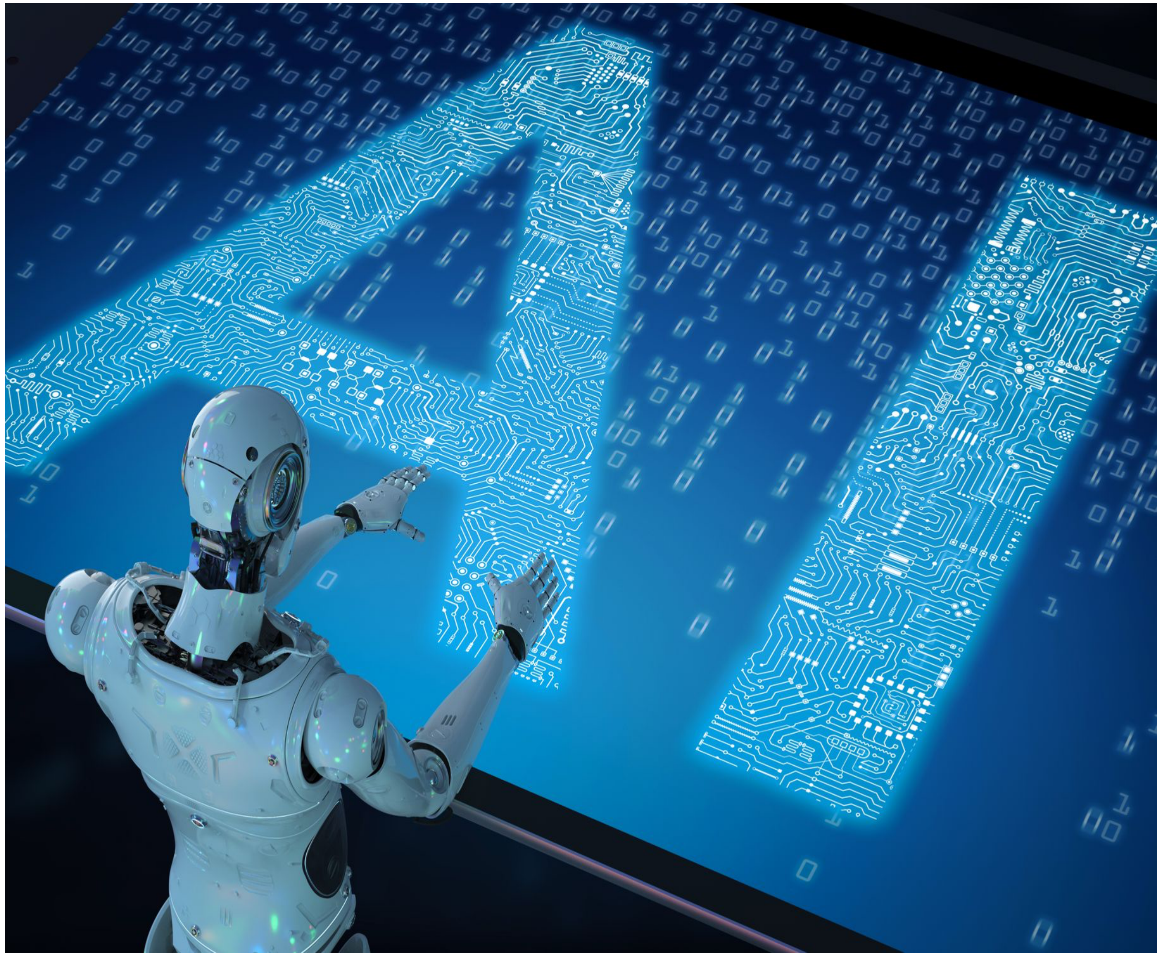
Autonomy in AI:
Merging Reasoning and Learning in Autonomous Agents

Giuseppe De Giacomo

University of Rome "La Sapienza", Italy

Autonomy in AI

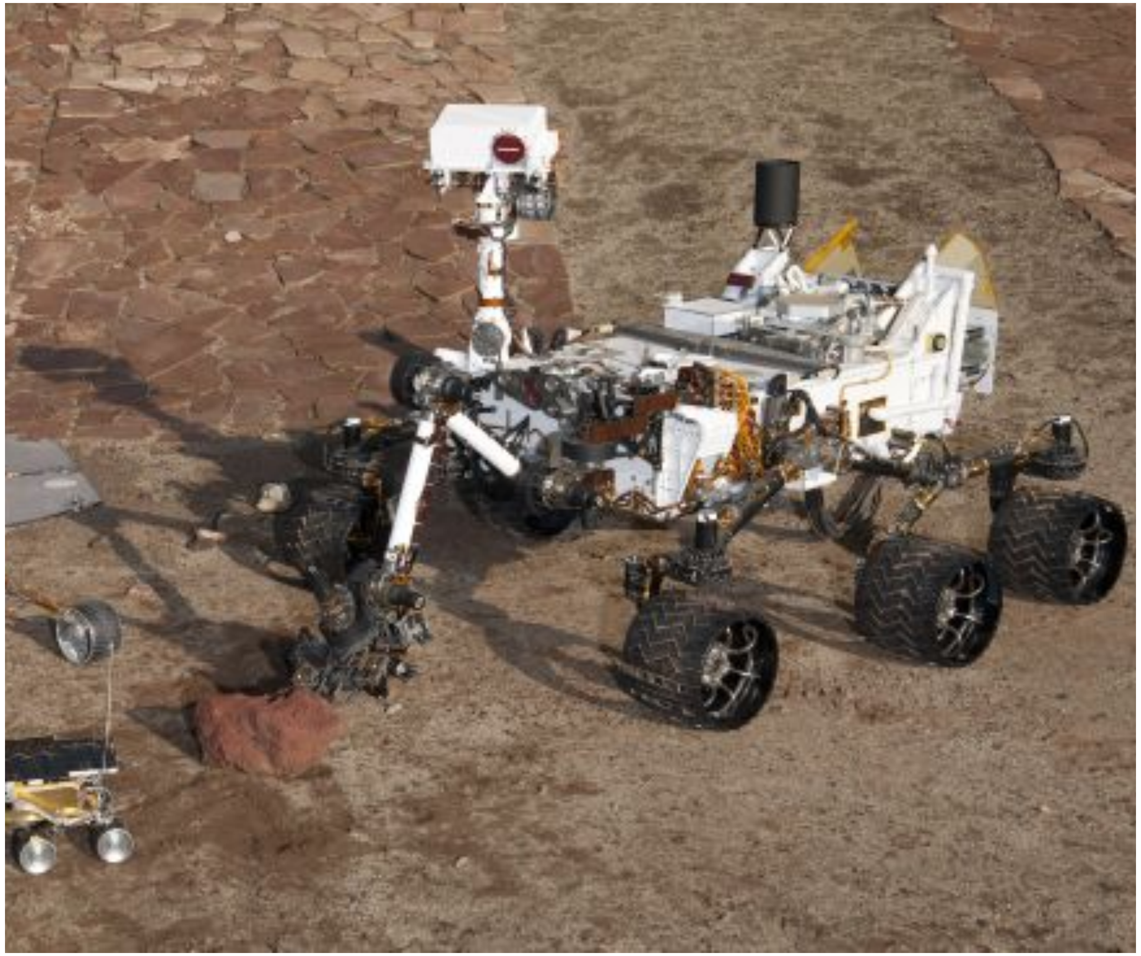
- Autonomy is one of the grand objectives of AI.
- Aims at building autonomous agents/robots that operate in changing, incompletely known, unpredictable environments.
- Requires autonomous reasoning and planning capabilities, as well as learning from experience.



Autonomy in AI: Space Exploration

Delay in communication requires high-level of autonomy during the mission.

Planning and scheduling for temporal extended goals is a top research topic at NASA.



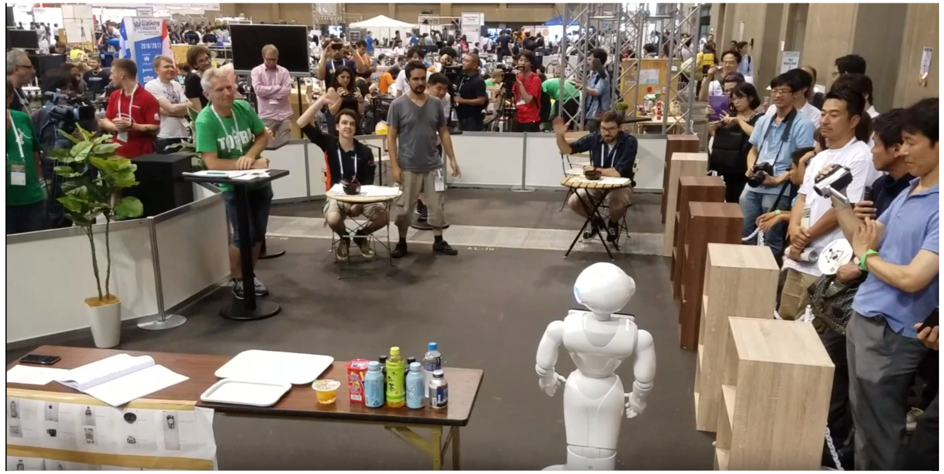
Autonomy in AI: Multi-Robot Systems in Logistics

Complex multi-robot systems need highly synchronized behaviours to fulfil their job.

These robots need autonomously resolve unexpected clashes.

Autonomy in AI: Interacting with Humans

Robots interacting with humans needs to properly schedule execution of complex actions that extends over time, as well as react to unexpected circumstances.



SAPIENZA UNIVERSITÀ DI ROMA

Autonomy Requires Reasoning and Learning

- Autonomy requires:
  - reasoning and planning capabilities
  - learning from experience

- Many areas of AI are concern with autonomy:
  - Logics in AI
  - Knowledge representation and reasoning
  - Planning
  - Multi-agent systems
  - Sequential decision making (MDPs)
  - Reinforcement learning

- Recently: some objectives are shared with automated synthesis in formal methods

WhiteMech: Whitebox Self Programming Mechanisms

ERC Advanced Grant

European Research Council
Established by the European Commission

SAPIENZA UNIVERSITÀ DI ROMA

Formal Methods

- Rigorous guarantees about the behavior of computational systems

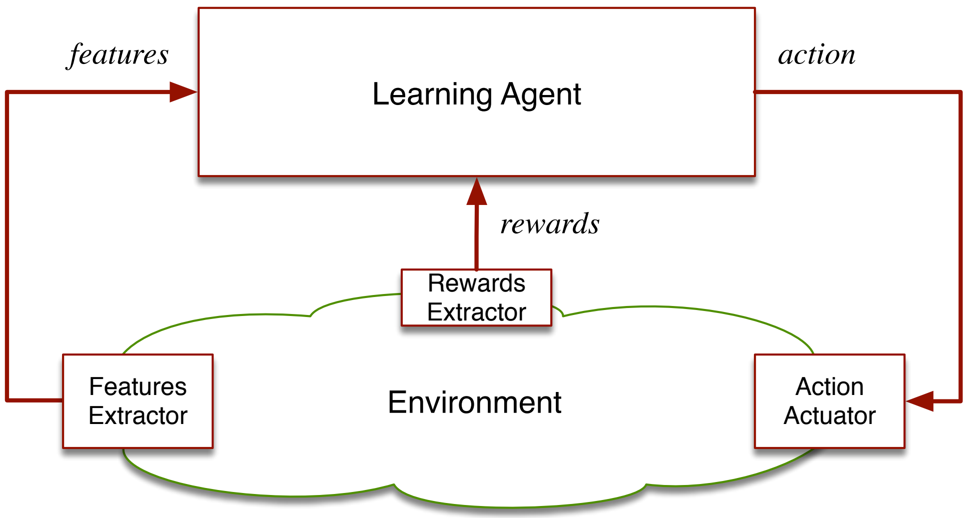
- Wide-spread industrial adoption

- Main tasks
  - Verification: given a system and a specification, check if the system satisfies the specification.
  - Automated program synthesis given a specification, synthesize a system that satisfies it.

Learning Agents and Reasoning Agents

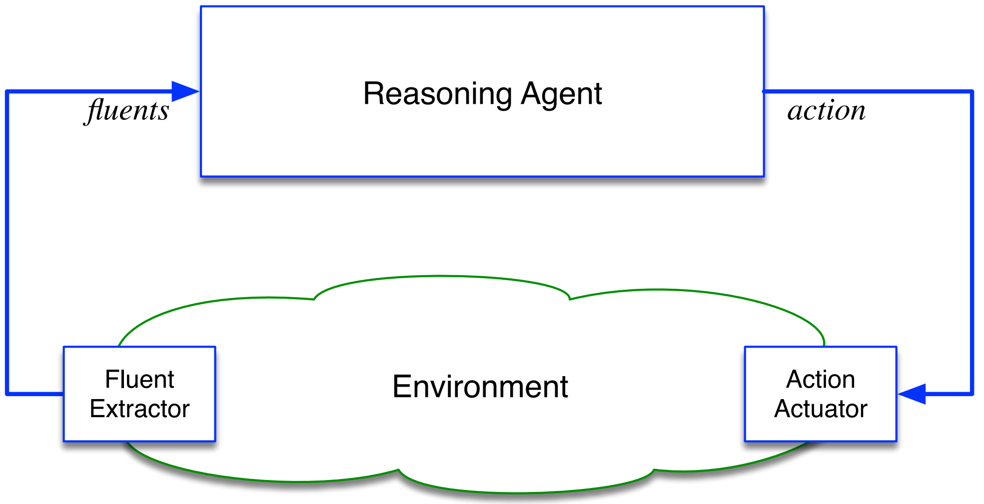
Learning agent:

- Senses and acts on the environment
- Gets rewards when right
- Does reinforcement learning



Reasoning agent:

- Senses and acts on the environment
- Has models of its environment and tasks
- Does reasoning and planning

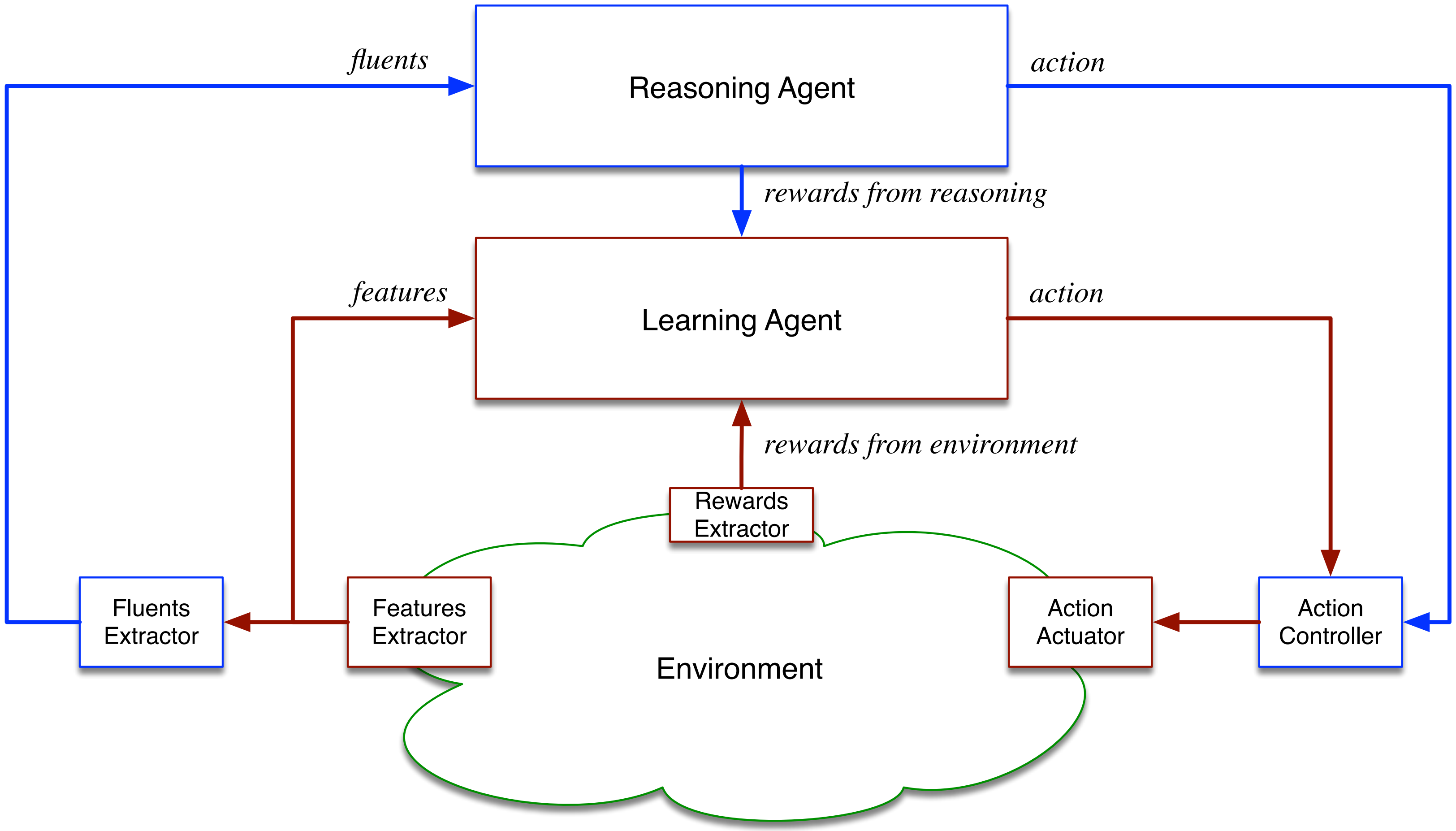


Merging Learning and Reasoning

Merging:

- Learning agent
  - Does reinforcement learning
  - Possibly deep reinforcement learning

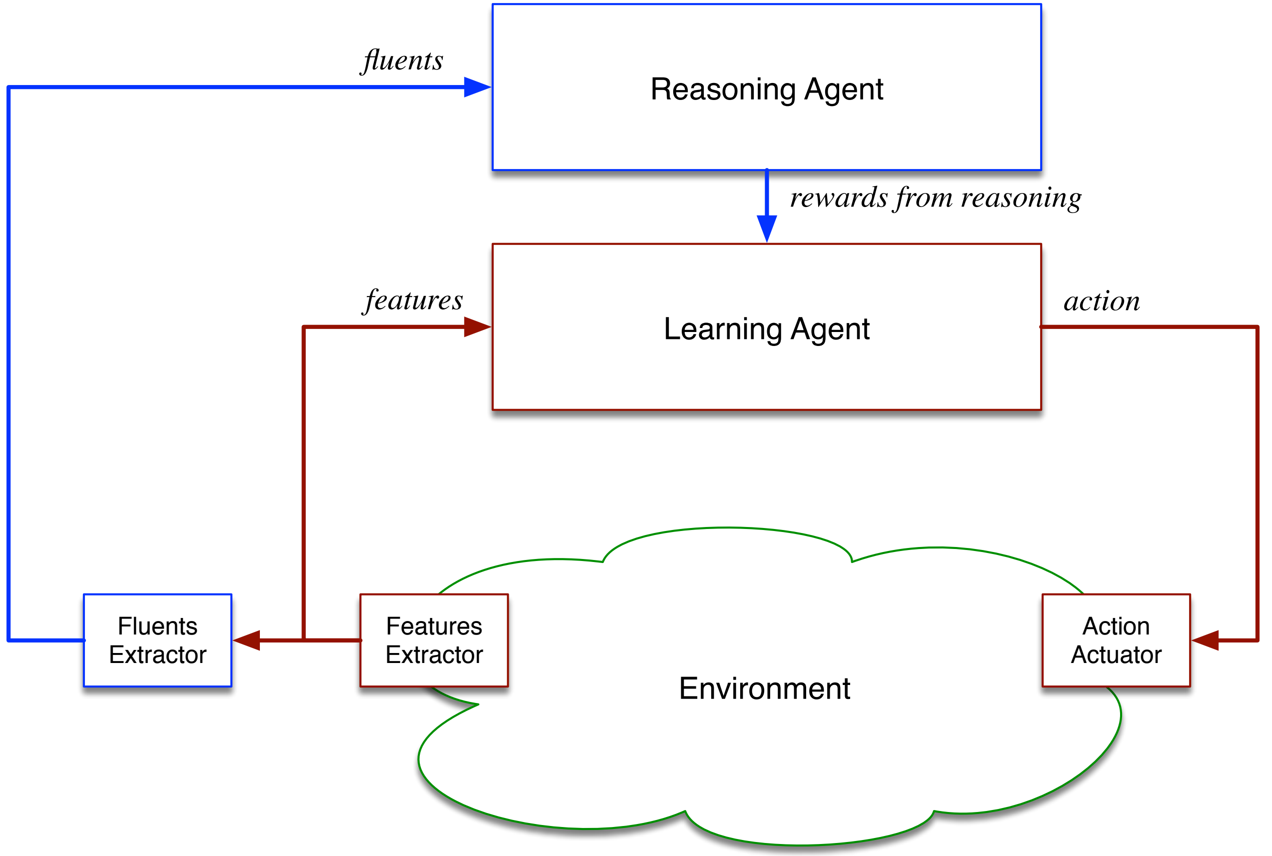
- Reasoning agent
  - Does reasoning
  - Possibly on temporal specification as in formal methods



MDPs with Logic-based non-Markovian Rewards

Simple Case:
MDPs with non-Markovian rewards

- Learning agent
  - Reinforcement Learning
  - Possibly deep reinforcement learning
- Reasoning agent
  - Rewards based on temporal logics
  - Improves rewards engineering



M. Littman. Programming agents via rewards. (Invited talk) IJCAI 2015.

R. Brafman, G. De Giacomo, F. Patrizi. LTLf /LDLf non-Markovian rewards. AAAI 2018.

A. Camacho, R. Icarte, T. Klassen, R. Valenzano, S. McIlraith. LTL and Beyond: Formal Languages for Reward Spec. in RL. IJCAI 2019.

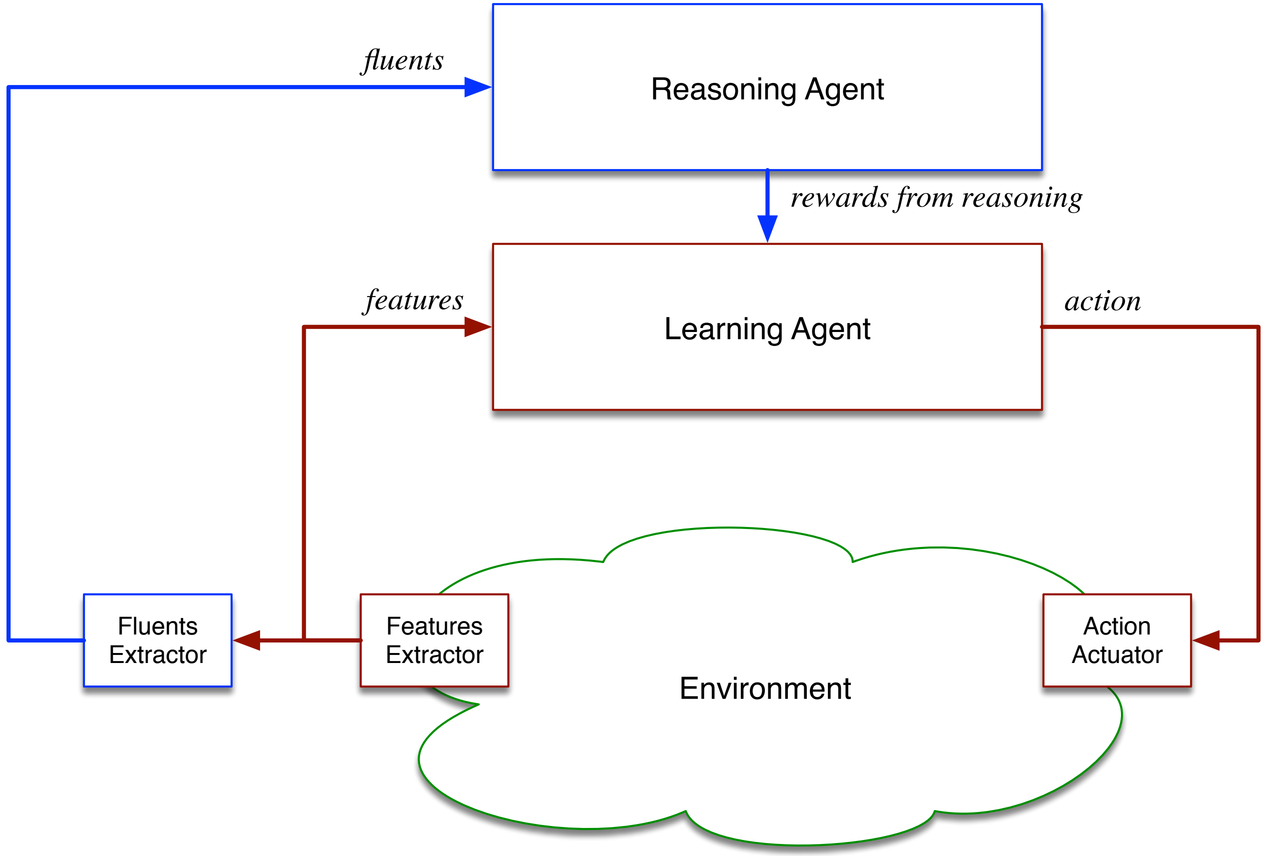
MDPs with Logic-based non-Markovian Rewards

Building blocks:

- Classic Reinforcement Learning
- Non-Markovian reward obtained from satisfaction of temporal formulas
- Temporal logics on finite traces
  - Focus on finite traces because agent's tasks terminate
  - LTLf a variants of LTL used in Formal Methods
  - LDLf extends LTLf with regular expressions for procedural constraints
  - Reasoning: transform

    LTLf/LDLf formula $\varphi \Rightarrow$ DFA A_φ

    for every trace τ: $\tau \models \varphi$ iff $\tau \in \ell(A_\varphi)$



M. Littman. Programming agents via rewards. (Invited talk) IJCAI 2015.

R. Brafman, G. De Giacomo, F. Patrizi. LTLf /LDLf non-Markovian rewards. AAAI 2018.

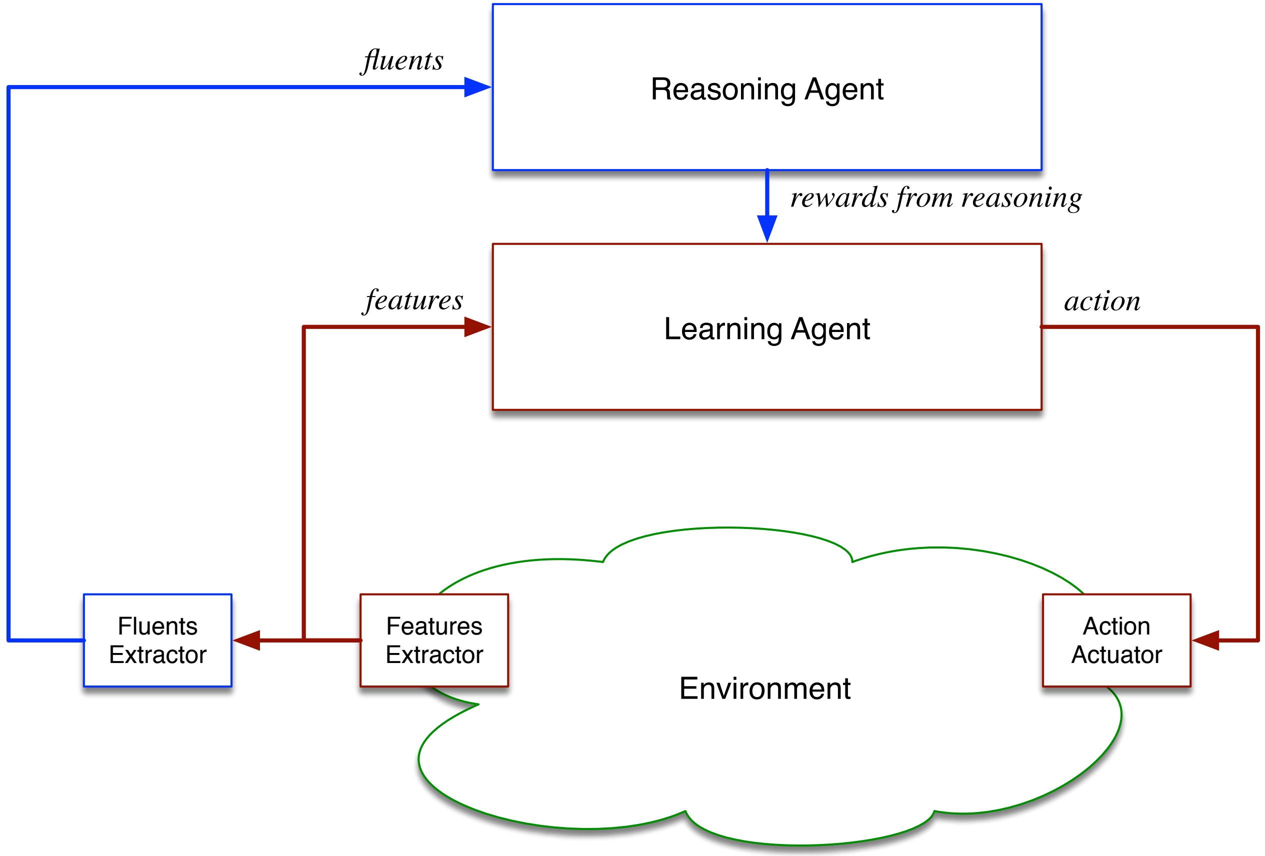
A. Camacho, R. Icarte, T. Klassen, R. Valenzano, S. McIlraith. LTL and Beyond: Formal Languages for Reward Spec. in RL. IJCAI 2019.

MDPs with Logic-based non-Markovian Rewards

Simple Case:
MDPs with non-Markovian rewards

- Learning agent: $\mathcal{M} = (S_{ag}, A_{ag}, Tr_{ag}, \cancel{R_{ag}})$
  MDP without rewards
- Reasoning agent: $\mathcal{R} = (\mathcal{L}, \{(\varphi_i, r_i)\}_{i=1}^m)$
  φ_i in LTLf/LDLf ⟹ $\overline{R}_{ag} : (S_{ag}, A_{ag})^* \to \mathbb{R}$
  non-Markovian rewards!
- Mapping between S_{ag} and $\mathcal{L}$

We can define equivalent MDP over an extended state space and do standard RL

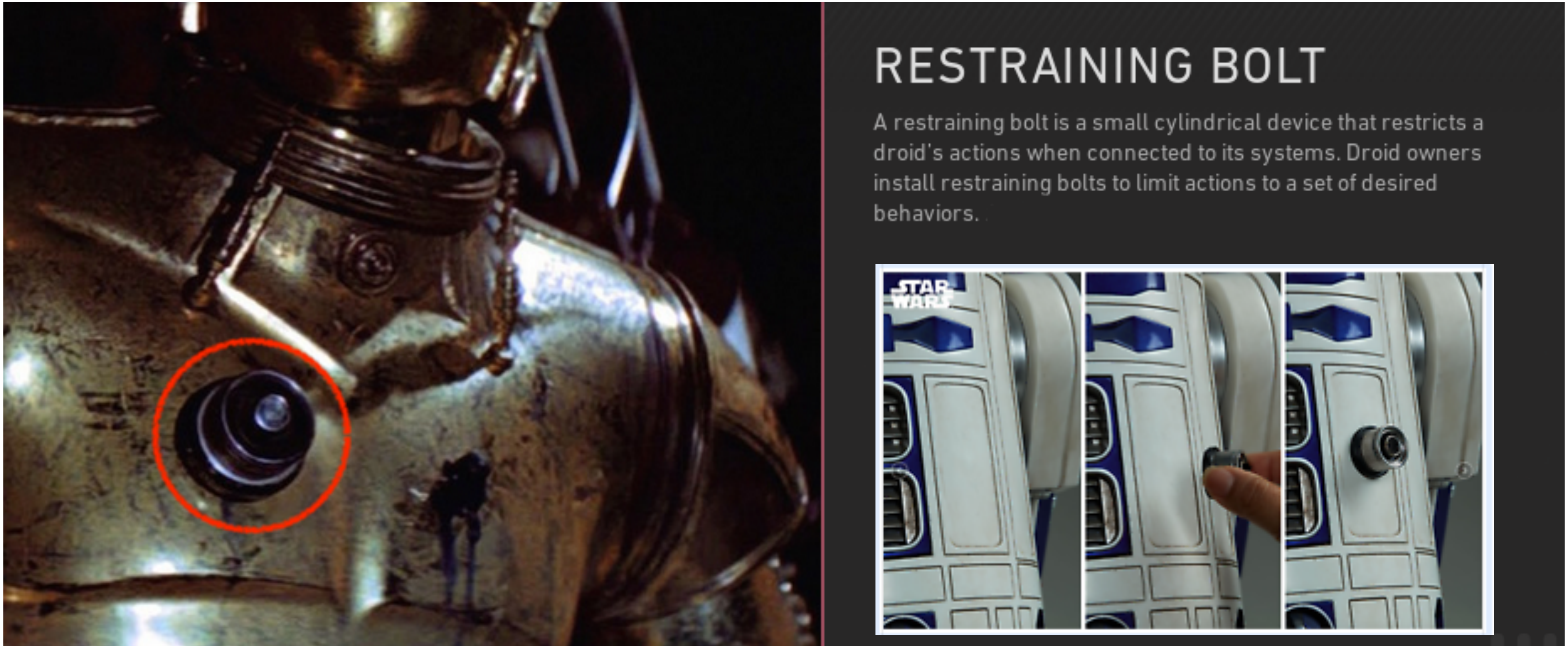


M. Littman. Programming agents via rewards. (Invited talk) IJCAI 2015.

R. Brafman, G. De Giacomo, F. Patrizi. LTLf /LDLf non-Markovian rewards. AAAI 2018.

A. Camacho, R. Icarte, T. Klassen, R. Valenzano, S. McIlraith. LTL and Beyond: Formal Languages for Reward Spec. in RL. IJCAI 2019.

Restraining Bolts



https://www.starwars.com/databank/restraining-bolt

Restraining Bolts

Two distinct representations of the world

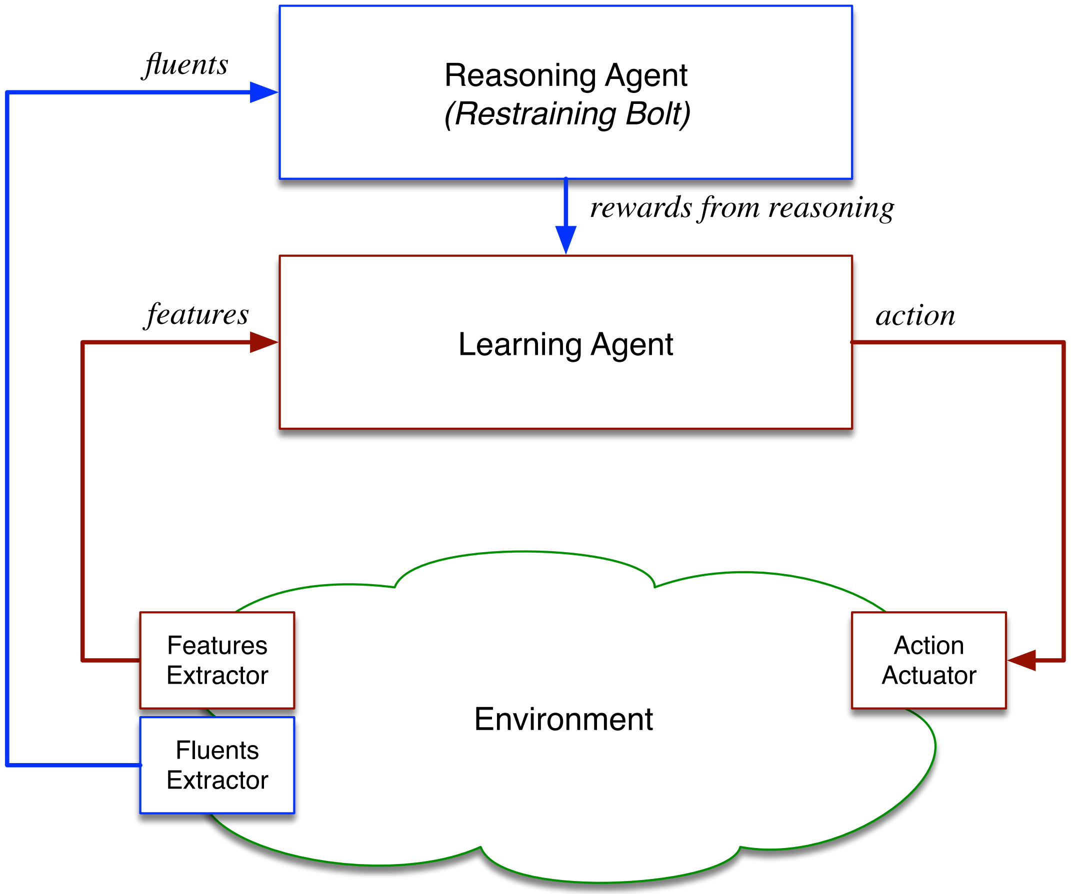
One for the agent,
by the designer of the agent

One for the restraining bolt,
by the authority imposing it

Restraining Bolts as Reasoning Agents

Double state representation (restraining bolts)

- Learning agent
  - Reinforcement learning
  - Possibly deep reinforcement learning
  - Uses features
- Reasoning agent (restraining bolt)
  - Rewards based on temporal logics
  - Improves rewards engineering
  - Uses human understandable fluents



G. De Giacomo, M. Favorito, L. Iocchi, and F. Patrizi. Foundations for Restraining Bolts: Reinforcement Learning with LTLf/LDLf Restraining Specifications. ICAPS 2019.

Restraining Bolts as Reasoning Agents

Double state representation (restraining bolts)

- Learning agent: $\mathcal{M} = (S_{ag}, A_{ag}, Tr_{ag}, \cancel{R_{ag}})$

  MDP without rewards

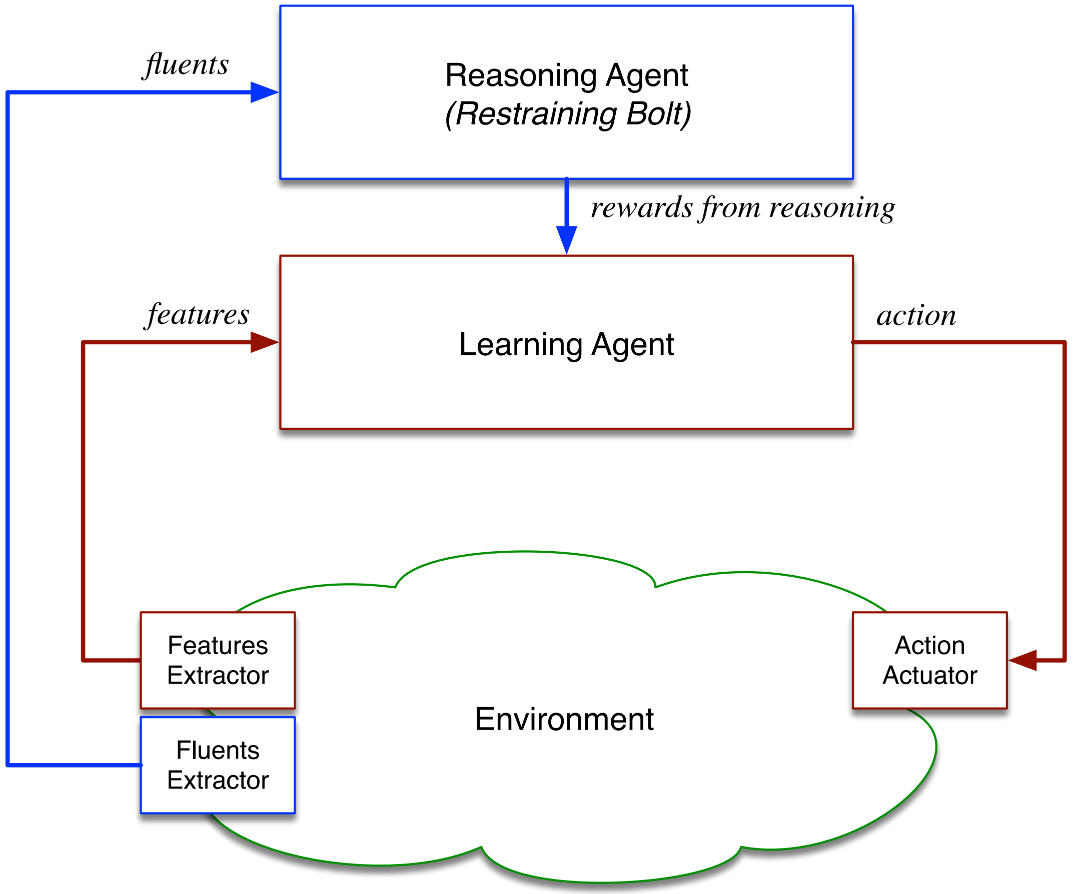
- Reasoning agent: $\mathcal{R} = (\mathcal{L}, \{(\varphi_i, r_i)\}_{i=1}^m)$

  φ_i in LTLf/LDLf ⇒ $\overline{R}_{ag} : (S_{ag}, A_{ag})^* \to \mathbb{R}$

  non-Markovian rewards!

- ~~Mapping between S_{ag} and $\mathcal{L}$~~

We can define equivalent MDP over an extended state space and do standard RL



G. De Giacomo, M. Favorito, L. Iocchi, and F. Patrizi. Foundations for Restraining Bolts: Reinforcement Learning with LTLf/LDLf Restraining Specifications. ICAPS 2019.

Restraining Bolts as Reasoning Agents

We can define equivalent MDP over an extended state space and do standard reinforcement learning

RL with $\mathrm{LTL}_f/\mathrm{LDL}_f$ restraining specifications for learning agent $M = \langle S, A, Tr_{ag}, R_{ag}\rangle$ and restraining bolt $RB = \langle \mathcal{L}, \{(\varphi_i, r_i)\}_{i=1}^m\rangle$

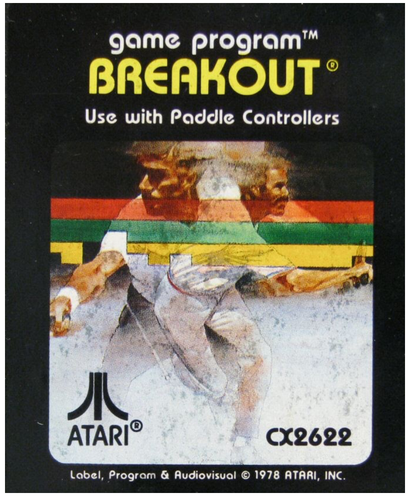
- **Transform each φ_i into DFA $\mathcal{A}_{\varphi_i} = \langle 2^{\mathcal{L}}, Q_i, q_{io}, \delta_i, F_i\rangle$** over fluents evaluations $\mathcal{L}$ with states Q_i and final states $F_i \subseteq Q_i$.
- **Do classical RL over a new MDP $M' = \langle Q_1 \times \cdots \times Q_m \times S, A, Tr'_{ag}, R'_{ag}\rangle$**
- **Thm: the optimal policy ρ'_{ag} learned for M' is an optimal policy of the original problem.**[a]

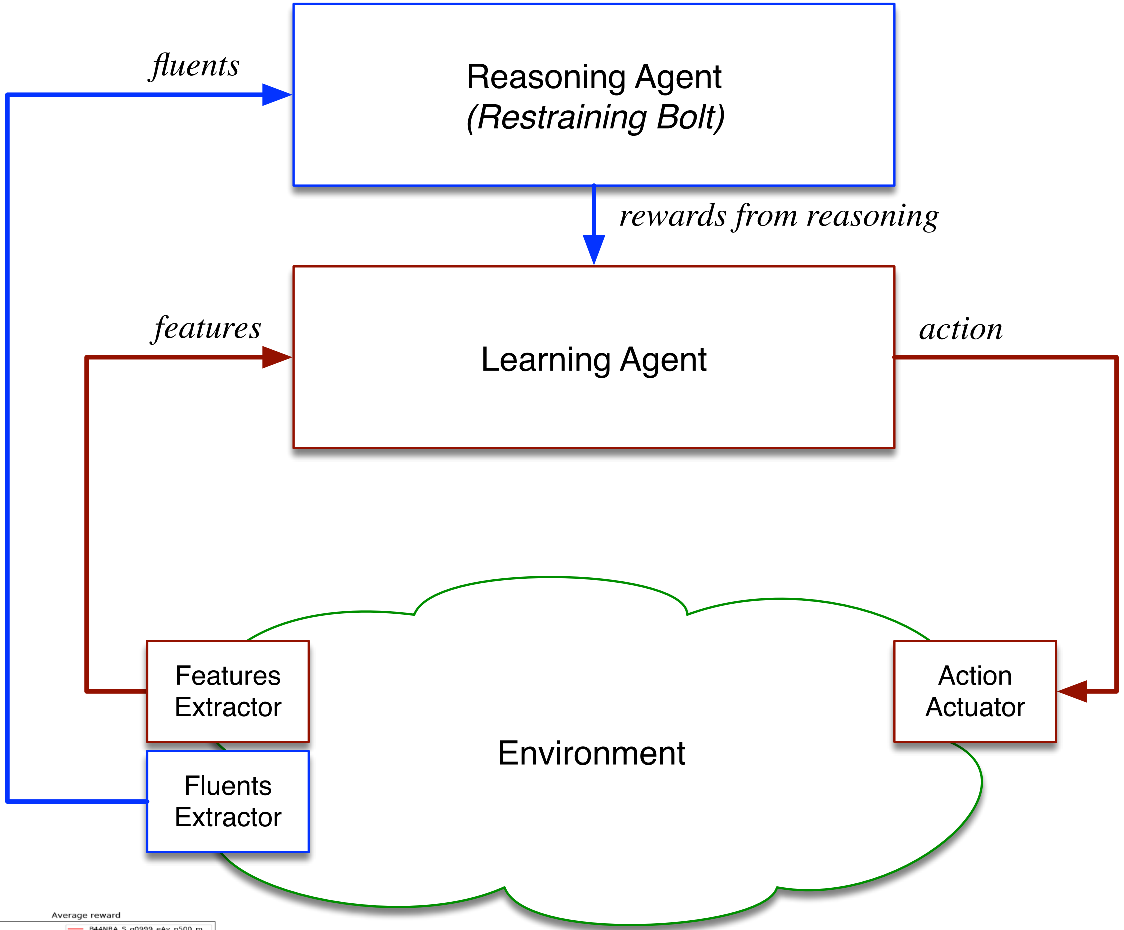
[a] Crux of the result: the reward function depends on features and automata states, not on fluents

$$R'_{ag}(q_1, \ldots, q_m, s, a, q'_1, \ldots, q'_m, s') = \sum_{i:q'_i \in F_i} r_i + R_{ag}(s, a, s')$$

G. De Giacomo, M. Favorito, L. Iocchi, and F. Patrizi. Foundations for Restraining Bolts: Reinforcement Learning with LTLf/LDLf Restraining Specifications. ICAPS 2019.

Example: Breakout



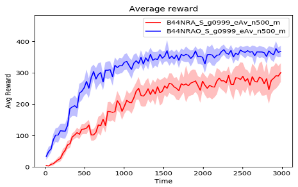


Learning Agent

- Features: paddle position, ball speed/position
- Actions: move the paddle
- ~~Rewards: reward when a brick is hit~~

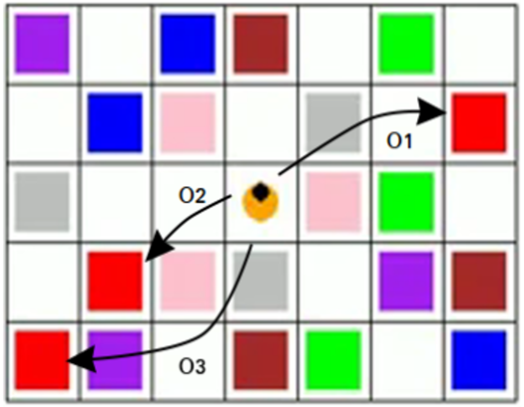
Restraining Bolt (Reasoning Agent)

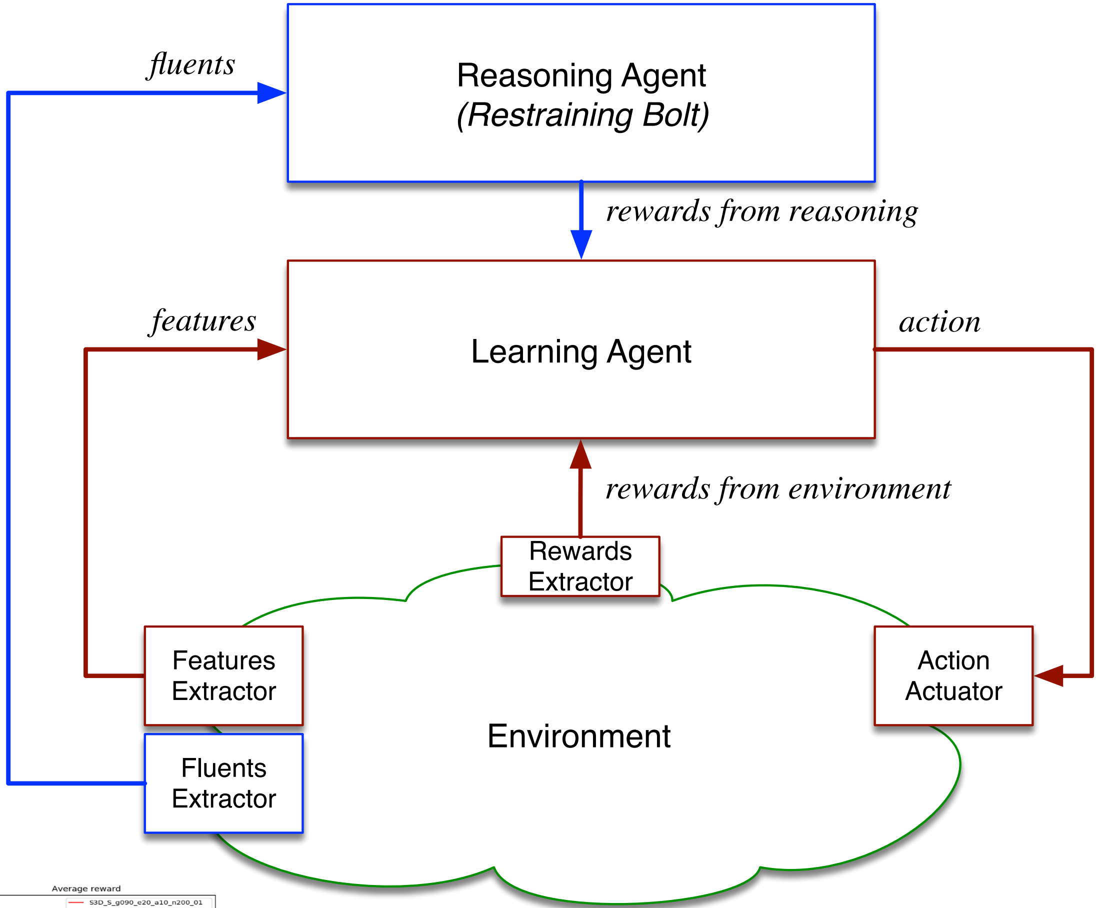
- Rewards: break one column at the time left to right (all bricks in column i must be removed before completing any other column j > i)
- Fluents: bricks/columns status (broken/not broken)



https://sites.google.com/diag.uniroma1.it/restraining-bolt

Example: Sapientino



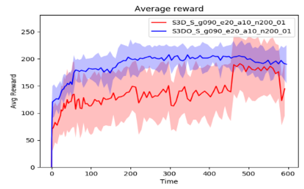


Learning Agent

- Features: robot position (x, y) and facing θ
- Actions: forward, backward, turn left, turn right, beep
- Rewards: negative rewards when agent exits board

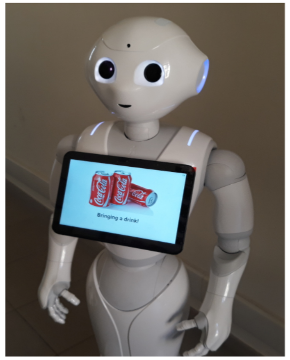
Restraining Bolt (Reasoning Agent)

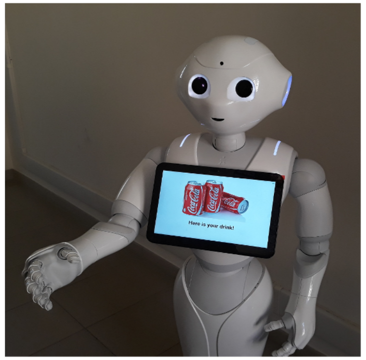
- Rewards: visit (beep) at a sequence of three positions of the same colour for each colour
- Fluents: colour of current cell, just beeped

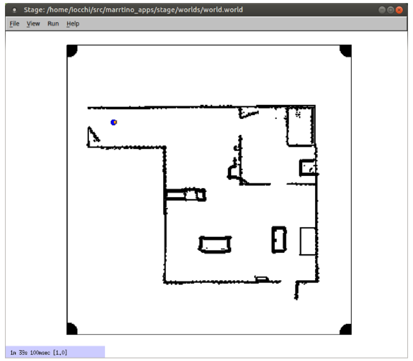


https://sites.google.com/diag.uniroma1.it/restraining-bolt

Example: Cocktail Party





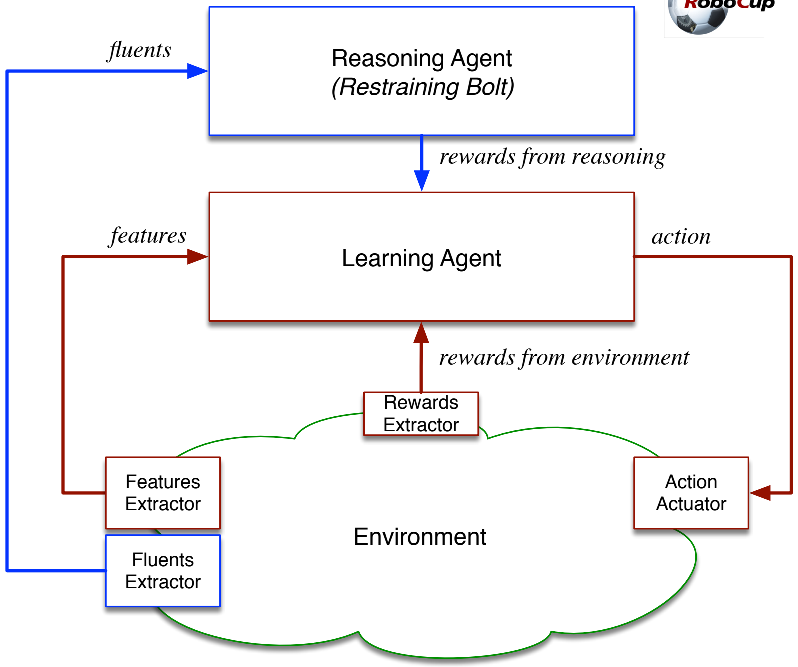


Learning Agent

- Features: robot's pose, location of objects (drinks and snacks), and location of people
- Actions: move in the environment, can grasp and deliver items to people
- Rewards: robot's navigation, deliver task is completed.

Restraining Bolt (Reasoning Agent)

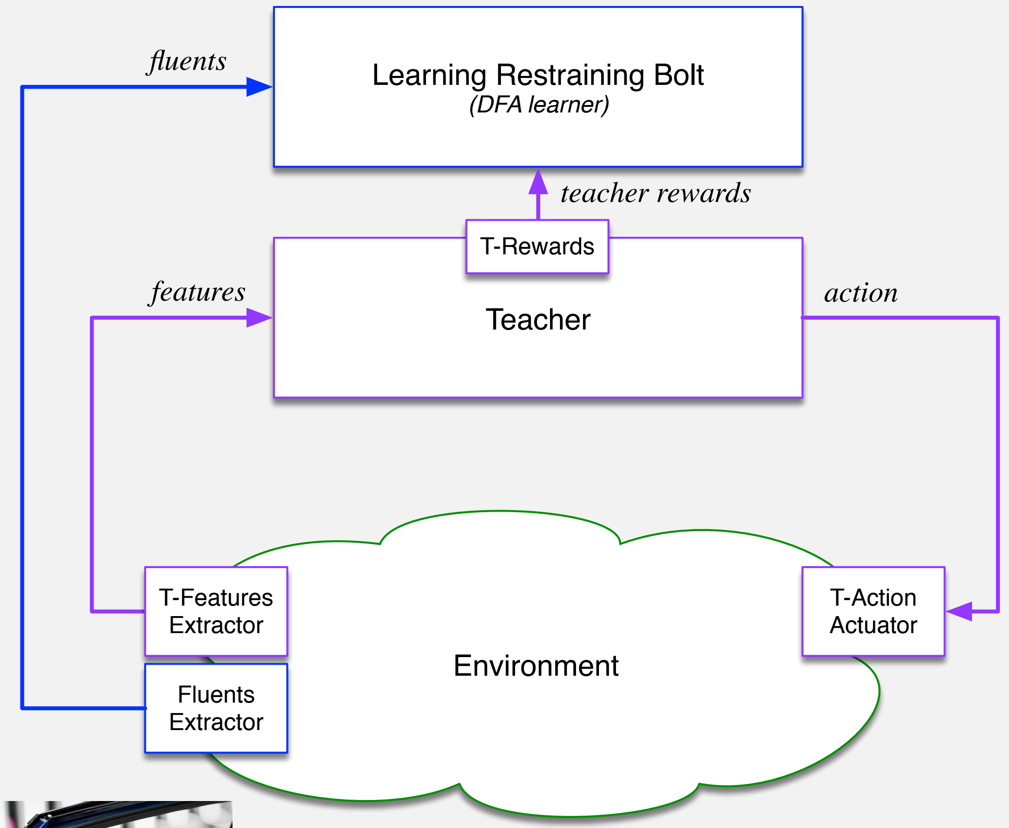
- Rewards: serve exactly one drink and one snack to every person, and do not serve alcoholic drinks to minors
- Fluents: identity and age of people, and received items *(uses Microsoft Cognitive Services Face API to provide information)*



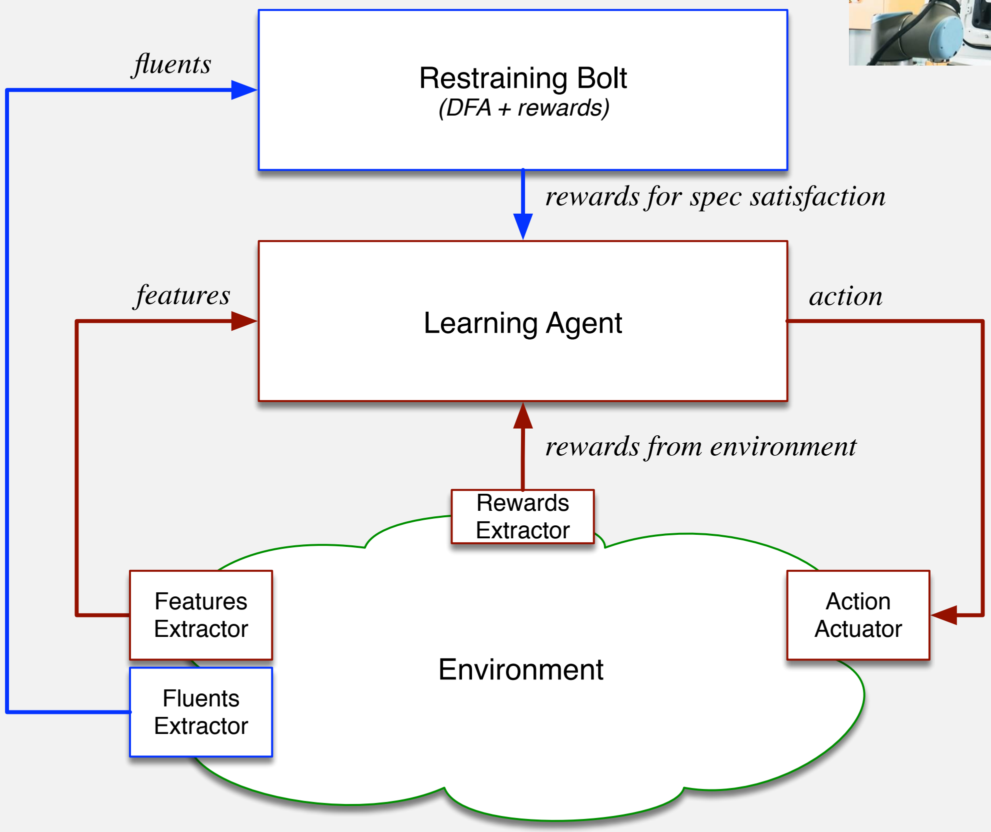
https://sites.google.com/diag.uniroma1.it/restraining-bolt

Extensions: Imitation Learning

Learn a Restraining Bolt



Use the Restraining Bolt



G. De Giacomo, M. Favorito, L. Iocchi, and F. Patrizi.
Imitation Learning over Heterogeneous Agents with Restraining Bolts. ICAPS 2020.
https://whitemech.github.io/Imitation-Learning-over-Heterogeneous-Agents-with-Restraining-Bolts

Conclusions

- Autonomy is one of the grand objectives of AI

- Important advancements from synergies among different areas of AI and CS:
  - Knowledge representation and reasoning
  - Planning
  - Multi-agents systems
  - Sequential decision making (MDPs)
  - Reinforcement learning
  - Formal methods

- Merging reasoning and learning is one of the most important challenges for autonomy in AI. Encouraging results are available

- One more thing. Goal Formation (where do the goal come from?) related to Goal Reasoning: obedient agents, rebellious agents, and agents that change mind through interaction.

G. De Giacomo, Y. Lesperance. Goal Formation through Interaction in the Situation Calculus: A Formal Account Grounded in Behavioral Science. AAMAS 2020. *(talk available on underline.io)*