AI Planning: a Key Component for Intelligent and Autonomous Systems

Damien Pellier
Damien.Pellier@imag.fr
http://lig-membres.imag.fr/pellier/

July 4, 2022

Laboratoire d'Informatique de Grenoble – Marvin Team
Université Grenoble Alpes

Part I

Introduction and Overview

Outline of Introduction

I. First Intuitions on Planning

Beating Lee Sedol is great, but...

⇒ Lee Sedol, 9th dan professional, one of the best Go players in the world when he played against the AlphaGo program in March 2016.

Beating Lee Sedol is great, but...

⇒ Lee Sedol, 9th dan professional, one of the best Go players in the world when he played against the AlphaGo program in March 2016.

- AlphaGo may be the best at playing Go, but what else can it do?

Beating Lee Sedol is great, but...

⇒ Lee Sedol, 9th dan professional, one of the best Go players in the world when he played against the AlphaGo program in March 2016.

- AlphaGo may be the best at playing Go, but what else can it do?
- Is that "intelligent system"?

Beating Lee Sedol is great, but...

⇒ Lee Sedol, 9th dan professional, one of the best Go players in the world when he played against the AlphaGo program in March 2016.

- AlphaGo may be the best at playing Go, but what else can it do?
- Is that "intelligent system"?
- How do you design systems capable of automatically solving new problems for which they were not specifically designed?

A First Intuitive Planning Definition

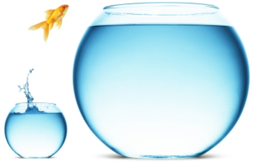
What is Planning ?

- Planning is the reasoning side of acting. It is an abstract, explicite deliberation process that chooses and organizes action by anticipating their outcomes.
- This deliberation aims at achieving as best as possible some pretated objectives.
- AI planning is an area of Artificial Intelligence (AI) that studies this deliberation process computationally to develop automous and intelligent systems.

AI Planning Ambitions (1/2)

Ambition: Write one program that can solve any problem.

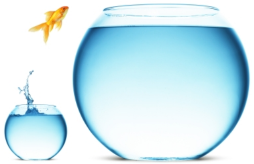
- Write $X \in \{\text{algorithms}\}$ such as $\forall\ Y \in \{\text{"problems"}\}$, X "solves" Y



AI Planning Ambitions (1/2)

Ambition: Write one program that can solve any problem.

- Write $X \in \{\text{algorithms}\}$ such as $\forall\ Y \in \{\text{"problems"}\}$, X "solves" Y
- What is a "problem"? What does it mean to "solve" it?

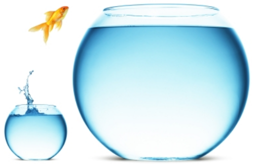


AI Planning Ambitions (1/2)

Ambition: Write one program that can solve any problem.

- Write $X \in \{\text{algorithms}\}$ such as $\forall\ Y \in \{\text{"problems"}\}$, X "solves" Y
- What is a "problem"? What does it mean to "solve" it?

In a more concrete and realistic way:



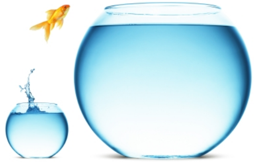
AI Planning Ambitions (1/2)

Ambition: Write one program that can solve any problem.

- Write $X \in \{\text{algorithms}\}$ such as $\forall\ Y \in \{\text{"problems"}\}$, X "solves" Y
- What is a "problem"? What does it mean to "solve" it?

In a more concrete and realistic way:

- **Ambition 1.** Write one program that can solve a large class of problems



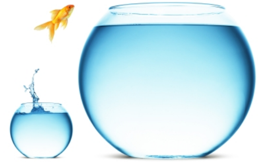
AI Planning Ambitions (1/2)

Ambition: Write one program that can solve any problem.

- Write $X \in \{\text{algorithms}\}$ such as $\forall\ Y \in \{\text{"problems"}\}$, X "solves" Y
- What is a "problem"? What does it mean to "solve" it?

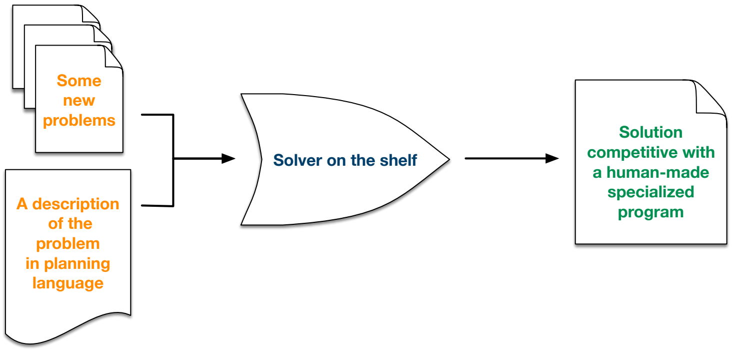
In a more concrete and realistic way:

- **Ambition 1.** Write one program that can solve a large class of problems
- **Ambition 2.** Write one program that can solve a large class of problems effectively
  - at least as effectively as a program developed specifically to solve these class of problems or as a human



AI Planning Ambition (2/2)

Ambition: Write one program that can solve all search problem.



AI Planning and General Problem Solving

- AI Planning = GPS (General Problem Solving)
- Two classical approaches of GPS:
  1. The blackbox description of a problem is an API (a programming interface) providing functionality allowing to construct the state space: InitialState(), GoalTest(s), etc.
     - ⇒ "Specifying the problem" = programming the API
  2. The declarative description of a problem comes in a problem description language. This allows to implement the API, and much more.
     - ⇒ "Specifying the problem" = writing a problem description.

⇒ For AI Planning, "problem description language" = planning language.

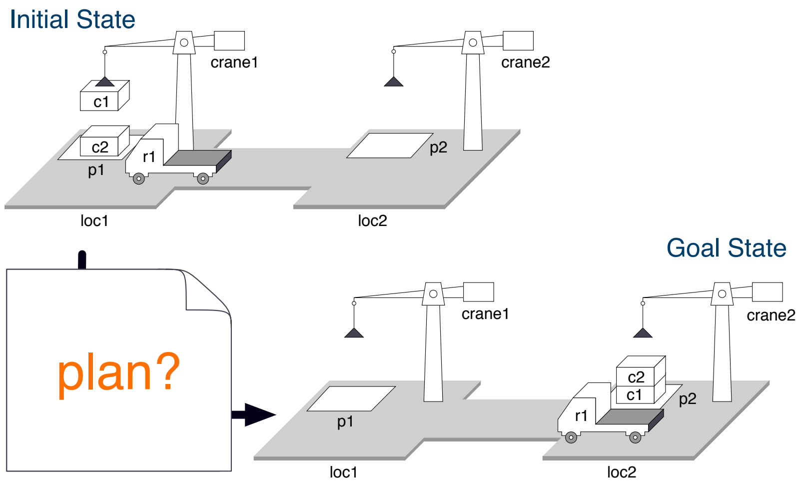
II. AI Planning Problem Definition

How does a planning language describe a problem?

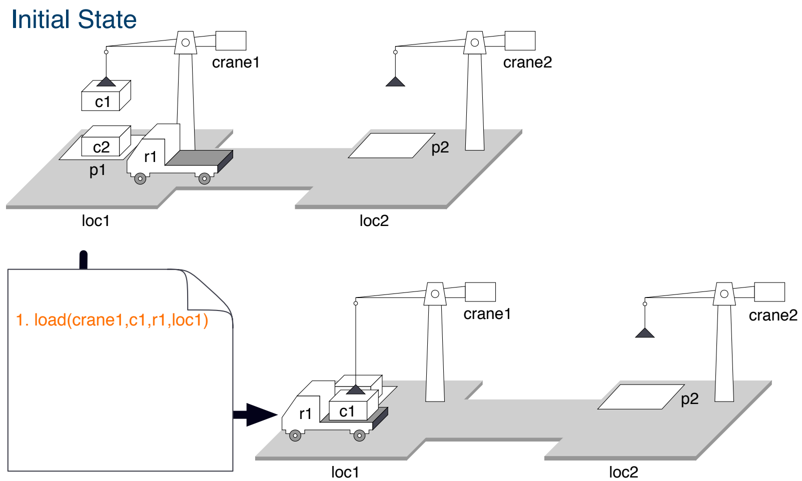
- A *logical description* of the possible states, e.g.,
  - holding(x,y), at(x, y), etc.
- A *logical description* of the initial state I, e.g.,
  - holding(robot, container), at(robot, dock), etc.
- A *logical description* of the goal condition G, e.g.,
  - delivered(container, dock), etc.
- A *logical description* of the set A of actions in terms of preconditions and effects, e.g.,
  - unload x, y, z:
    pre : holding(x,z) $\wedge$ at(x, z)
    eff : $\neg$holding(x,z) $\wedge$ delivered(y, z)

$\Rightarrow$ Solution (plan) = sequence of actions from A, transforming I into a state that satisfies G, e.g., "unload robot, container, dock"

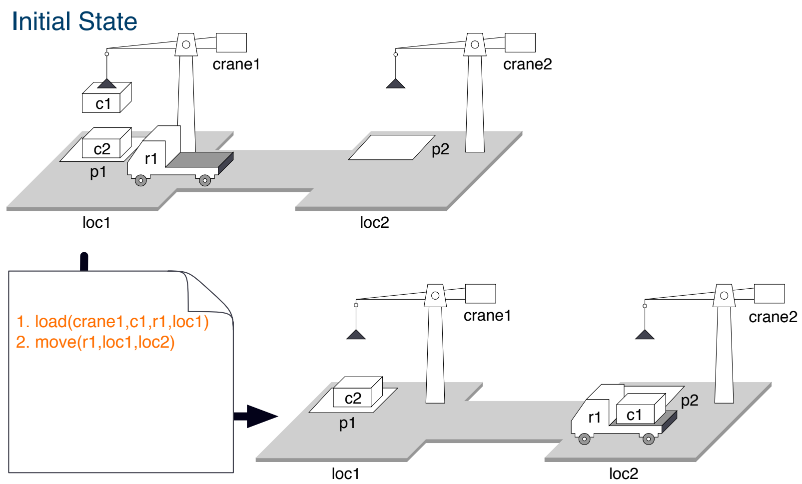
Example: the robot dockers (1/3)



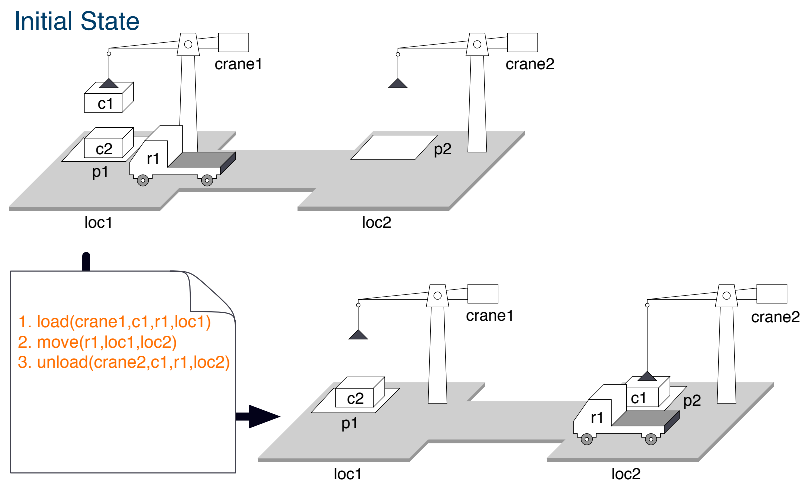
Example: the robot dockers (1/3)



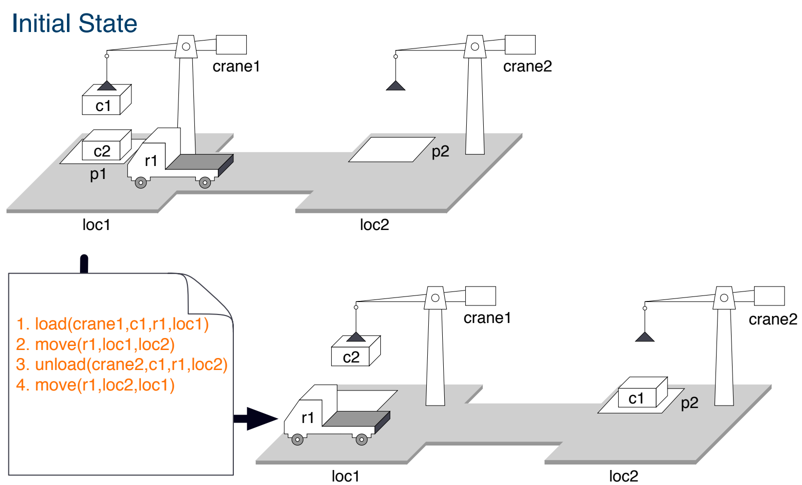
Example: the robot dockers (1/3)



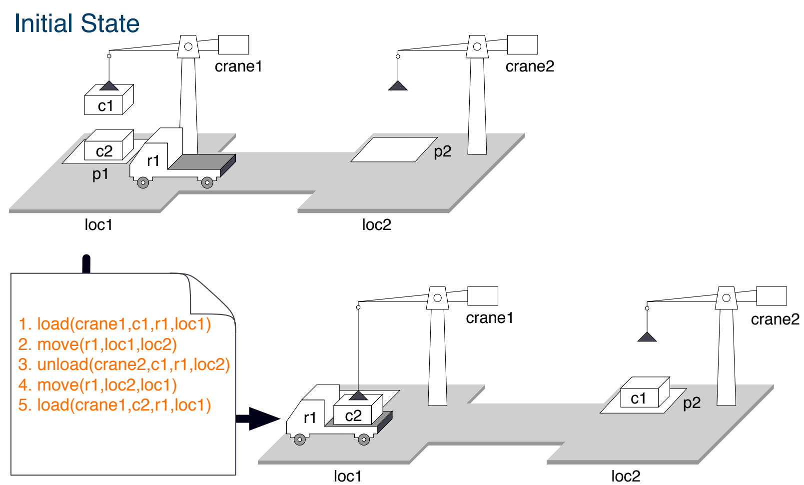
Example: the robot dockers (1/3)



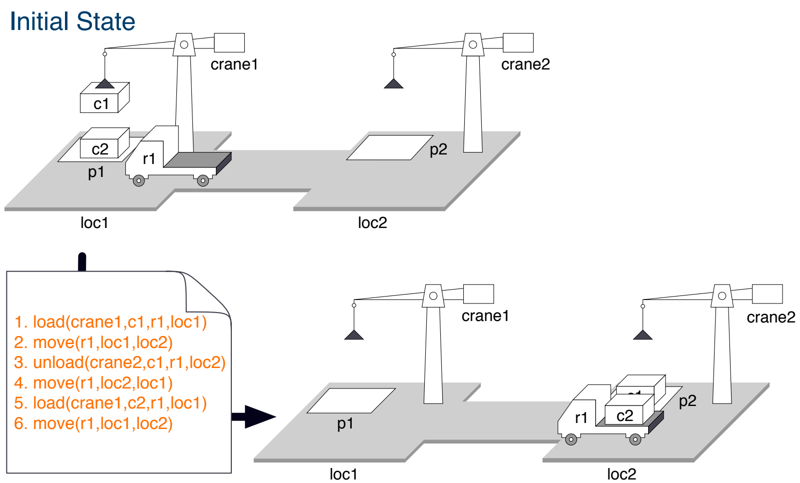
Example: the robot dockers (1/3)



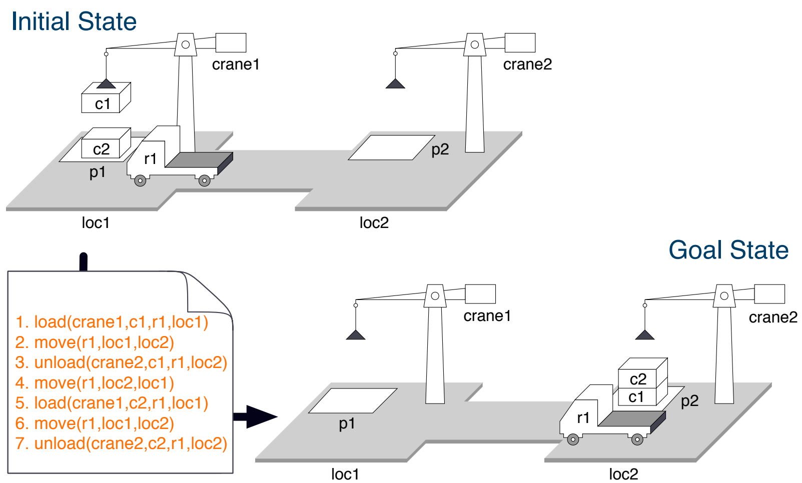
Example: the robot dockers (1/3)



Example: the robot dockers (1/3)



Example: the robot dockers (1/3)



Example: the robot dockers (2/3)

Domain PDDL

```
(define (domain dwr)
  (:requirements :strips :typing)
  (:types location pile robot crane container)
  (:predicates
   (adjacent ?l1 ?l2 - location)
   (at ?r - robot ?l - location)
   (occupied ?l location)
      ....
  (:action move
   :parameters (?r - robot ?from ?to - location)
   :precondition (and (at ?r ?from) (not (occupied ?to))
                      (adjacent ?from ?to))
   :effect (and (at ?r ?from)
                (not (occupied ?from)) (occupied ?to))
  )
  ...
```


Example: the robot dockers (3/3)

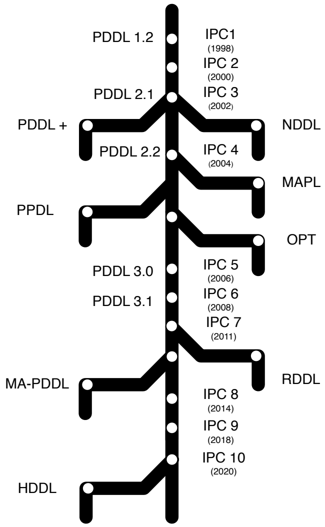
Problem PDDL

```
(define (problem pb)
 (:domain dwr)
 (:objects
     c1 c2 - container
     r1 - robot
     crane1 crane2 - crane
     loc1 loc2 - location
     p11 p12 p21 p22 - pile)
 (:init
  (adjacent loc1 loc2) (holding crane1 c1) (at r1 loc1)
  (attached p11 loc1) (belong crane1 loc1) ....
 )
 (:goal (and (in c2 p2) (on c1 c2)))
)
```

How expressive are the planning languages?

⇒ Several PDDL extensions have been proposed to express complex problems:

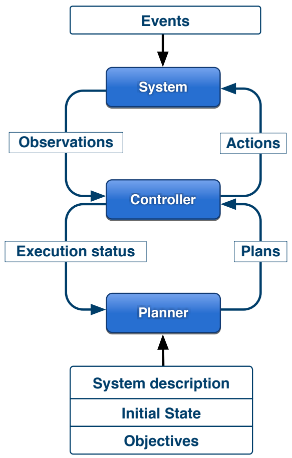
- Time constraints (PDDL2.1)
- Exogenous events and continuous changes (PDDL+)
- Muliagent problems (MAPLE, MA-PDDL)
- Onthologie (OPT)
- Probabilistic problems (PPDDL, RDDL)
- Complex contraints on plan trajectory (PDDL3.0 and 3.1)
- Hierarchical problem (HDDL)
- etc.



III. AI Planning Concepts

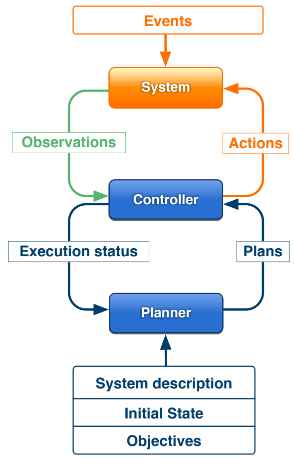
AI Planning Architecture

- **AI planning architecture** is classically depicted in through the interaction between three components



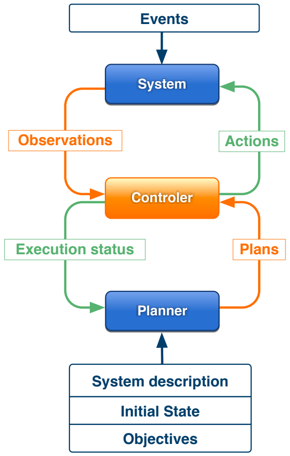
AI Planning Architecture

- **AI planning architecture** is classically depicted in through the interaction between three components
  1. A state transition system that evolves according to the events and actions receive.



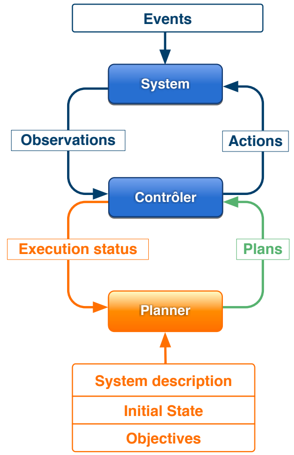
AI Planning Architecture

- **AI planning architecture** is classically depicted in through the interaction between three components
  1. A state transition system that evolves according to the events and actions receive.
  2. A controller, given as input the state of the system, provides as output an action according to some plan.



AI Planning Architecture

- **AI planning architecture** is classically depicted in through the interaction between three components
  1. A state transition system that evolves according to the events and actions receive.
  2. A controller, given as input the state of the system, provides as output an action according to some plan.
  3. A planner, given as input a description of a planning problem in a declarative manner (the actions that can be executed by the robot, an initial situation, and some objective), synthesizes a plan for the controller in order to achieve the objectives.



State Transition System

- A state transition system is formally defined as a 4-tuple $\Sigma = (S, A, E, \gamma)$, where:
  - $S = \{s_1, s_2, \ldots, s_n\}$ is a finite or recursively enumerable set of states
  - $A = \{a_1, a_2, \ldots, a_n\}$ is a finite or recursively enumerable set of actions
  - $E = \{e_1, e_2, \ldots, e_n\}$ is a finite or recursively enumerable set of events
  - $\gamma : S \times A \times E \rightarrow 2^S$ is a state transition function
- A state transition system may be represented by a directed graph whose nodes are the state in S.

Planning Objective

Planning Objective

Given a state transition Σ, the purpose planning is to find which set of organized actions (plan) to apply to which states in order to achieve some objective when starting from a given situation.

- A plan is a structure that gives the appropriate actions.
- The objective can be specified in several different ways:
  1. The simplest specification consists of a goals state.
  2. The objective can be also expressed by the satisfaction of some conditions over the sequence of state followed by the system.
  3. The objective can be expressed by an utility function attached to each states, with penalties and rewards.
  4. The objective can be expressed as a task that the system should perform.

AI Planning Model Assumptions

Classical **AI planning model makes** many simplifying **assumptions** :

- The state transition system that models the states of the world is finite, fully observable, deterministic and static
- The plans are sequential
- The notion of time is implicit, e.g. actions have no duration.
- The computation of a solution plan is centralized
- Planning does not reason on numeric functions
- *etc.*

AI Planning Model Assumptions

Classical **AI planning model makes** many simplifying **assumptions** :

- The state transition system that models the states of the world is finite, fully observable, deterministic and static
- The plans are sequential
- The notion of time is implicit, e.g. actions have no duration.
- The computation of a solution plan is centralized
- Planning does not reason on numeric functions
- *etc.*

Problems

1. Many real-world robotics applications require the removal of several of the listed assumptions.
2. Taking all these assumptions into account makes planning problems untractable in the general case and leads to a combinatorial explosion.

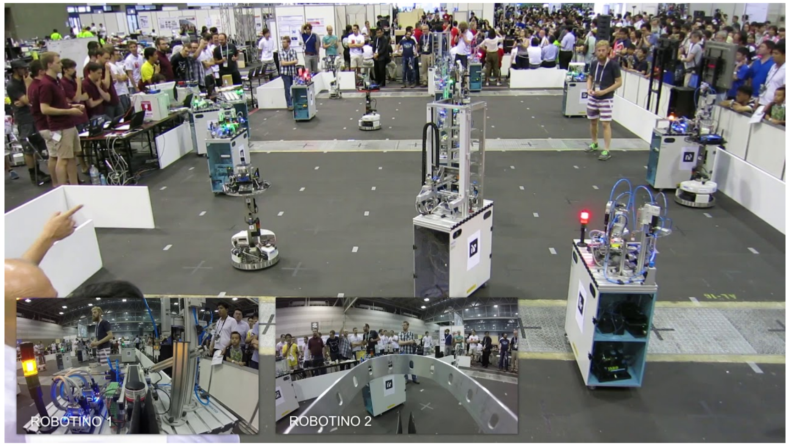
The main techniques used in AI planning

- The main techniques used in AI planning are:
  - State Space planning
  - Plan-Space planning
  - SAT and CSP planning
  - Hierarchical TaskNetwork Planning (HTN)
  - Planning based on Markov Decision Process
  - Planning based Model Checking
  - Planning based linear and integer programming
  - *etc.*

Scope of application

- Many target areas:
  - military, aerospace, industrial, everyday life, IT, etc.
- Examples of applications:
  - Video games
  - Intelligent building management
  - Road or air traffic management
  - Composition of web services
  - Industrial process optimization
  - Supervision of robotic systems, e.g., AGV, AUV
  - Warehouse logistics management
  - *etc.*

Exemple of Manufacturing Robotics Application



AI Planning Hot Scientific Challenges

- Explainable Planning
  - Generation and evaluation of explanations
  - Mixed-planning ...
- Hierarchical Planning
- Integrated Execution of Planning and Acting for Robotics
  - Mission, path, and motion planning for robots
  - Human-aware planning and execution in human-robot interaction ...
- Knowledge Engineering for Planning and Scheduling
  - Methods and tools for the acquisition of domain knowledge
  - Formal languages for domain description ...
- Heuristics and Search for Domain-independent Planning
- Tool for AI planning

IV. About the content of this course...

What we are tackled in this course

1. Classical approaches of planning
   - Classical Planning representation and complexity
   - State space planning
   - Plan space Planning
2. Neo-classical approaches of planning
   - Planning Graph techniques
   - Propositional Plannning techniques
   - Constraints satisfaction techniques
3. Heuristics and control strategies for planning
   - Hierarchical Task Network planning
   - Heuristics search for planning
4. Planning under uncertainly
   - Planning based on Markov Decision processes
   - Planning based on Model Checking

What we are not tackled in this course

- Other forms of planning:
  - Path and Motion Planning is concerned with the synthesis of a geometric path from a starting position in space to a goal for mobile systems, such as a truck, a mechanical arm, a robot, etc.
  - Perception Planning is concerned with plans involving senseing actions for gathering informations generally to model the environnement of a autonmous system.
  - Navigation Planning combines the two previous problems of motion and perception planning in order to reach a goal or to explore an area.
  - Manipulation Planning is concerned with handling objects, e.g., to build assemblies.

Course Evaluation and Exam

- The course is based on continuous assessment
- A few exercises will have been done during the course sessions to illustrate certain concepts.
- The TP sessions will be devoted to the development of planners and will be the subject of reports. The reports will be used to calculate the course mark.

Further readings

Bibliography

M. Ghallab, D. Nau and P. Traverso
Automated Planning Theory and Practice
Morgan Kaufmann, 2004

M. Ghallab, D. Nau and P. Traverso
Automated Planning and Acting
Morgan Kaufmann, 2018

P. Régnier and V. Vidal
Algorithmes de la planification en IA
Cépaduès, 2004