PMR-3510 Inteligência Artificial

Aula 16 - Developing Planners

Prof. José Reinaldo Silva
reinaldo@usp.br

d-l@b

Definition: A *STRIPS planning problem* consists of a STRIPS operator specification, a set Σ of initial propositions and a set Ω of goal propositions.

Definition: A *solution* to a STRIPS planning problem is a plan (sequence of operators) α such that, in every graph where the STRIPS operator specification holds, we have $\Sigma \rightarrow [\alpha]\Omega$.

Definition: A *STRIPS operator specification* consists of a set of *operator symbols* where each operator symbol is associated with a *prerequisite list*, an *add list* and a *delete list* each of which is a set of proposition symbols.

Definition: A STRIPS operator specification is said to *hold* (or be *valid*) in a graph search problem if for each operator o_i, and each node n such that every prerequisite of o_i is true at n, we have the following conditions.

- All propositions on the add list of o_i are true at the node $o_i(n)$.
- If P is a proposition that is *not* on the delete list of o_i, and P is true at n, then P is true at $o_i(n)$.

https://www.ibm.com/blogs/research/2020/02/automated-planning-aaai-2020/

IBM Products & Solutions Consulting & Services Learn & Support Explore more Search

IBM Research Blog Topics Labs About

AI

IBM Automated Planning Research @ AAAI 2020

February 4, 2020 | Written by: Tathagata Chakraborti, Michael Katz, and Shirin Sohrabi

Categorized: AI

Share this post:

What is Planning?

Planning refers the task of finding a procedural course of action for a declaratively described system to reach its goals while optimizing target performance measures. Given the task, automated planning technologies find a sequence of operators or a plan that can transition the current state to a desired end state.

Automated planning provides an autonomous agent with the ability to answer the question *"What should I do next?"* in order to function in its environment. This was among the first tasks posed in the early formulation of artificial intelligence more than half a century ago and — to this date — remains a largely unsolved core AI competency due to the complexity of the task in terms of both representation and reasoning.

Automated Planning at IBM

THE AI PLANNER OF THE FUTURE

AI-enabled decision-making in Supply Chains and Logistics

Prof.dr. Tom Van Woensel (Program leader)

The AI PLANNER OF THE FUTURE research program is hosted by the Department of Industrial Engineering & Innovation Sciences and is supported by the European Supply Chain Forum, Department of Industrial Engineering & Innovation Sciences, the Eindhoven Artificial Intelligence Systems Institute, the Logistics Community Brabant and the TKI Dinalog. The program connects to the different communities, moonshots strategic agendas and the themes of each of these supporting partners. It combines 25 researchers, 10 PhD students and over 50 Bachelor and Master students, for the coming five years (2021-2026). The research program is explained in detail below, including the 10 individual PhD projects.

Setting the scene

The Internet of Things enables the instant exchange of data and information between machines, operators, and organizations. As such, supply chains benefit from an instant exchange of information on inventory availability, supply conditions, etc. Corrective and scheduled maintenance becomes condition-based and predictable, while Digital Twins are used to further optimize production- and maintenance processes. All these evolutions allow to optimize the performance of the key operational processes eventually leading to improvements on both internal (i.e., efficient, and effective processes) and external objectives (i.e., customer value, competitive advantages).

However, although promising, transforming the traditional supply chains and logistics, and their operations (manufacturing, transportation, order fulfillment, inventory, etc.), to fully reap the benefits is challenging. That is, decision-making in Supply Chain and Logistics is complex and different from decision-making in other areas as it involves many intertwined multi-disciplinary decisions (think of transport, inventory, location, human resources, ICT systems, etc.) and key performance indicators (including people, profit, planet dimensions and also ethics and corporate social responsibility).

The quintessential dream of Artificial Intelligence (AI) for automated planning (in supply chain and logistics) is to create a world without any human planners. All planning operations and tasks, all feedback, all data is processed automatically by advanced algorithms. Thus, everything is done completely automatically, and autonomously, without the need for any human intervention. Along the same lines, McKinsey (2018) coined the notion of no-touch planning, aiming to eliminate the role of humans in planning as people only add error (as compared to a completely data-driven planning process). This line of reasoning often leads to popular “consultancy-like” quotes as:

The ESCF companies involved in the AI PLANNER OF THE FUTURE research program

AMPLEON, BAYER, ams, NEWAYS, Office DEPOT, Ewals Cargo Care

EASTMAN, Shell, DHL, سابك sabic, Nike, nexperia

ASML, NXP, HILTI, TRINSEO, BAUSCH Health, DOW

Amway, RHI MAGNESITA, TLN, Royal HaskoningDHV, HEINEKEN, HUAWEI

evofenedex, vidaXL, SATURN MediaMarkt, DB SCHENKER, DAF

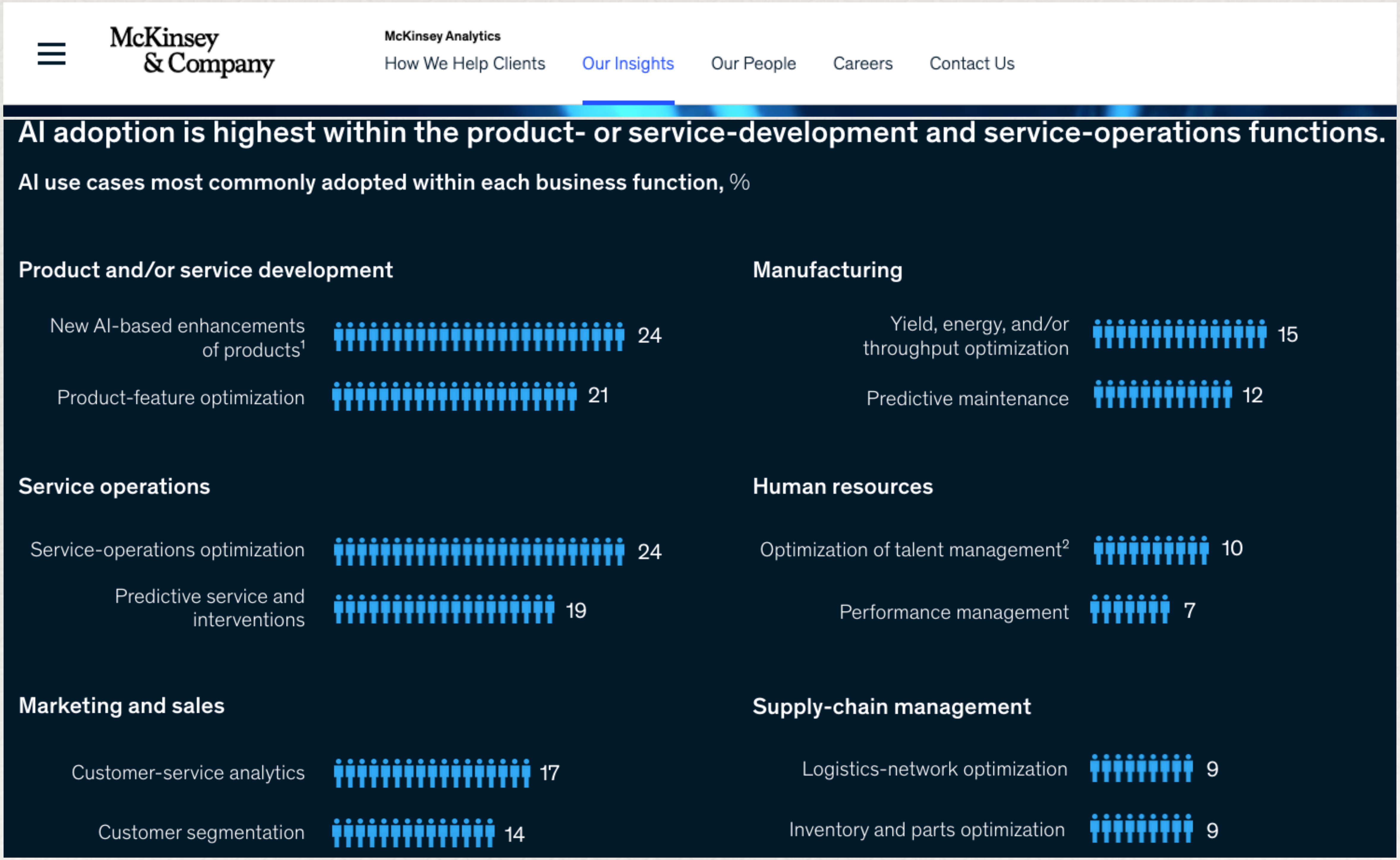
JUMBO, Royal Flora Holland, nabuurs supply chain solutions, CQM, Bonduelle Group, SACHEM

Container Centralen, SAP, SupplyStack, essity, ABInBev, VDL

MMGuide Business Intelligence, ORTEC, DE PERSGROEP NEDERLAND, LOUWMAN DEALER GROUP, Dr.Oetker, NTS GROUP

TKT, ICT smarter solutions, pipple, CTAC enabling your ambition, ThermoFisher SCIENTIFIC, Canon

DEN HARTOGH, H&S GROUP, PHILIPS, HUTCHISON PORTS ECT ROTTERDAM, eyeon years ahead https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/global-survey-the-state-of-ai-in-2020



The Knowledge Engineering Approach

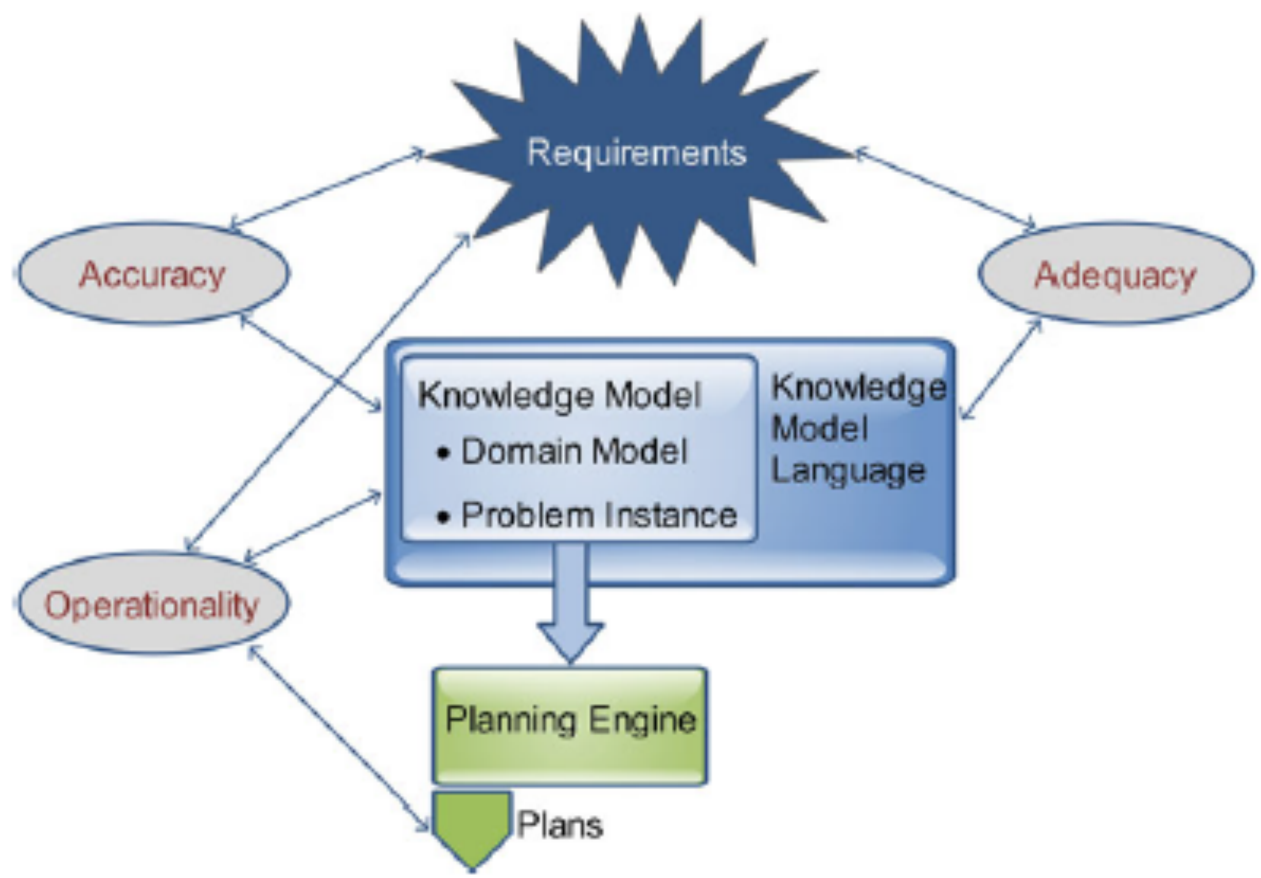


Fig. 3.1 PDM (planning domain modeling properties) basic properties as defined by LeeMc-Cluskey et al. and discussed later in McCluskey [27]

Home | Calls | Program | Registration | Committees | Attending | Code of Conduct

Dates

March 28, 2022 - Submissions Due

April 30, 2022 - Notification

May 15, 2022 - Camera-ready Due

June 15, 2022 - Workshop Date

Workshops

- HPlan
- PRL
- RDDPS

KEPS

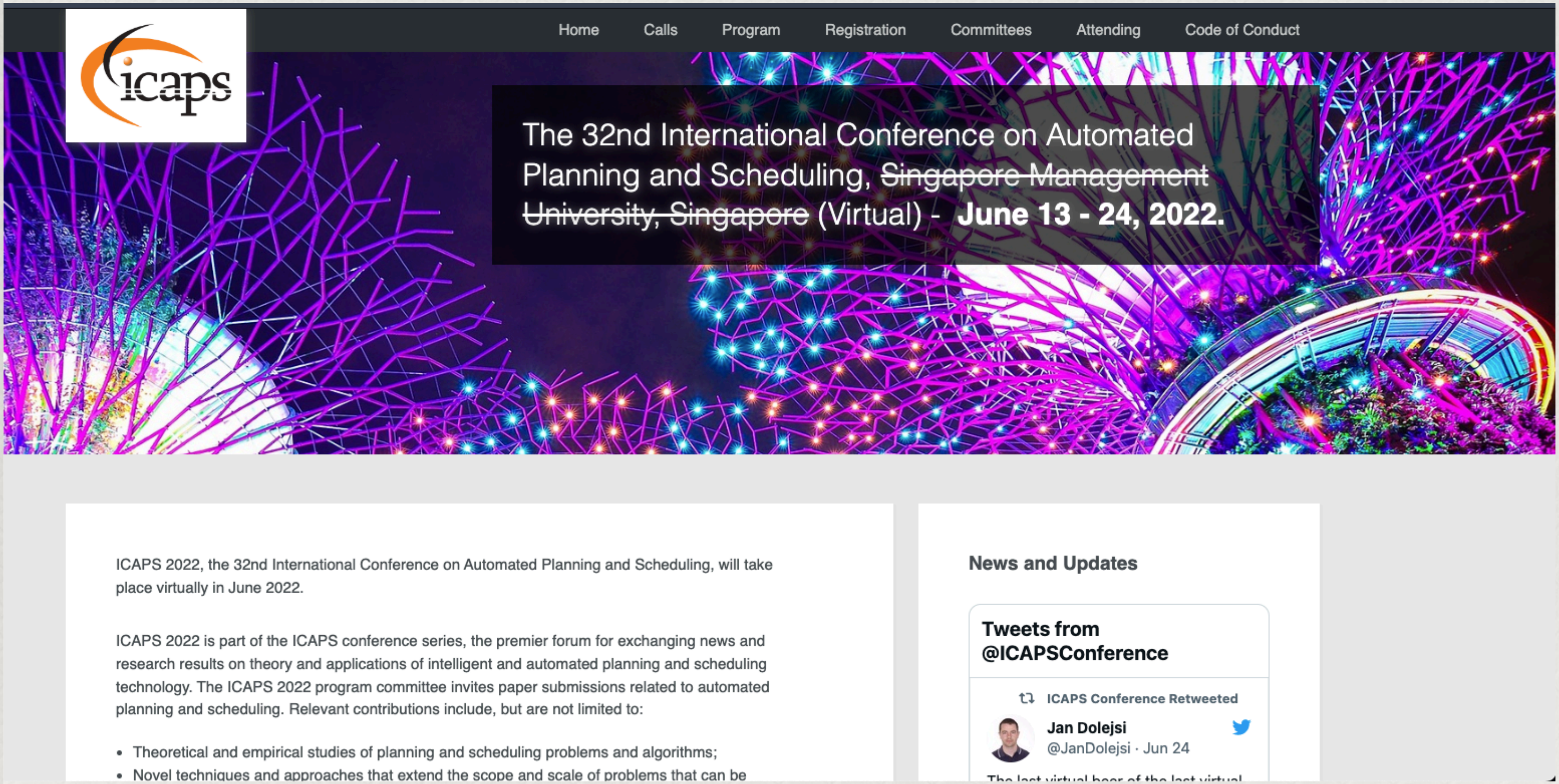
2022 Workshop on Knowledge Engineering for Planning and Scheduling

An ICAPS'22 Workshop
Singapore
15 June 2022

Despite the progress in automated planning and scheduling systems, these systems still need to be fed by carefully engineered domain and problem description and they need to be fine-tuned for particular domains and problems. Knowledge engineering for AI planning and scheduling deals with the acquisition, design, validation and maintenance of domain models, and the selection and optimization of appropriate machinery to work on them. These processes impact directly on the success of real-world planning and scheduling applications. The importance of knowledge engineering techniques is clearly demonstrated by a performance gap between domain-independent planners and planners exploiting domain dependent knowledge.

The workshop shall continue the tradition of several International Competitions on Knowledge Engineering for Planning and Scheduling (ICKEPS) and KEPS workshops. Rather than focusing only on software tools and domain encoding techniques –which are topics of ICKEPS– the workshop will cover all aspects of knowledge engineering for AI planning and scheduling.

Topics


Home
Calls
Program
Registration
Committees
Attending
Code of Conduct
icaps
The 32nd International Conference on Automated Planning and Scheduling, ~~Singapore Management University, Singapore~~ (Virtual) - June 13 - 24, 2022.
ICAPS 2022, the 32nd International Conference on Automated Planning and Scheduling, will take place virtually in June 2022.
ICAPS 2022 is part of the ICAPS conference series, the premier forum for exchanging news and research results on theory and applications of intelligent and automated planning and scheduling technology. The ICAPS 2022 program committee invites paper submissions related to automated planning and scheduling. Relevant contributions include, but are not limited to:
• Theoretical and empirical studies of planning and scheduling problems and algorithms;
• Novel techniques and approaches that extend the scope and scale of problems that can be
News and Updates
Tweets from @ICAPSConference
ICAPS Conference Retweeted
Jan Dolejsi
@JanDolejsi · Jun 24

icaps

Home Calls Program Registrat

Platinum Sponsors

INVITAE

JPMorgan Chase & Co.

Gold Sponsors

Gold Sponsors

IBM

SIFT

NISSAN

Silver Sponsors

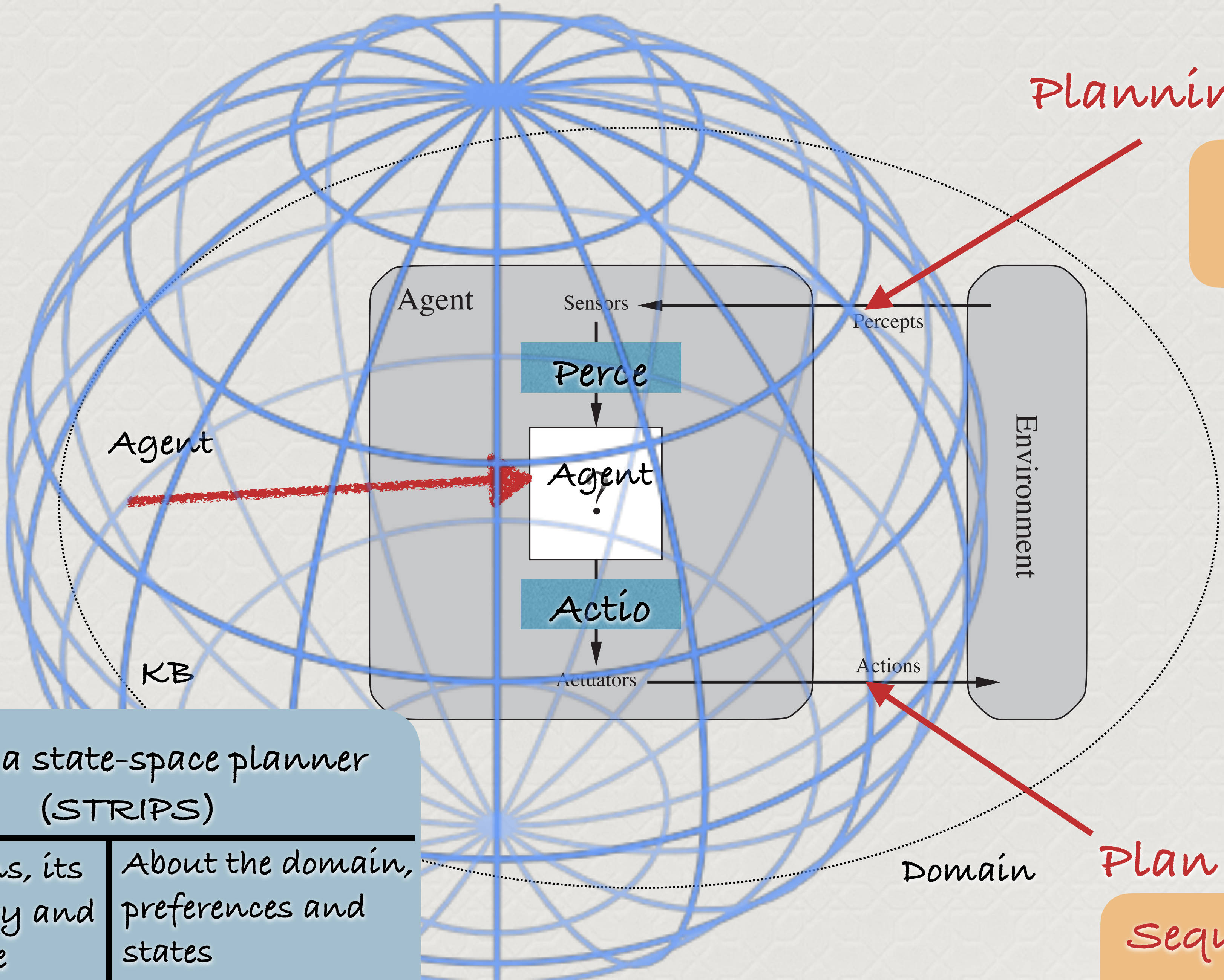
Google

Adventium Labs

Planning problem

Initial and final state

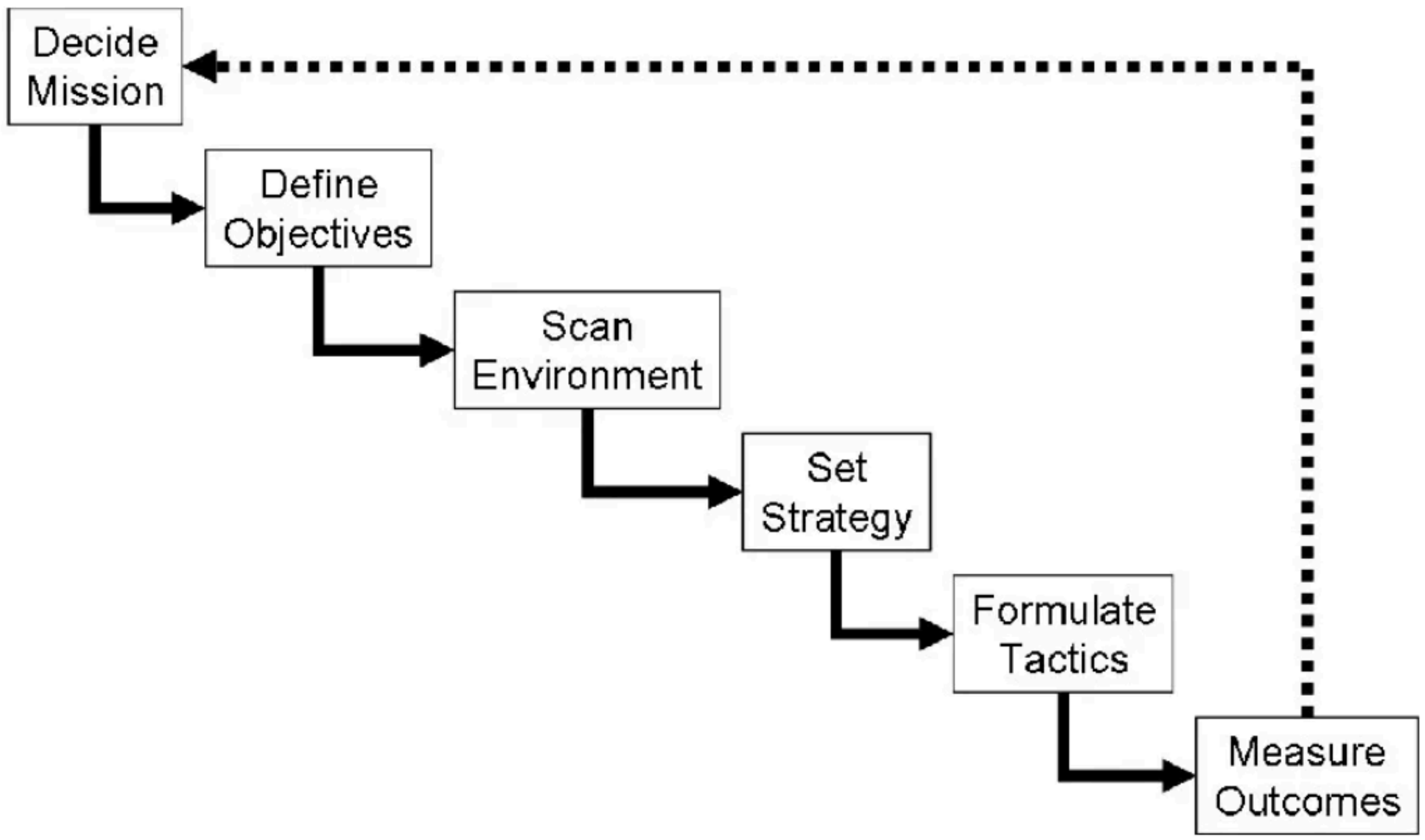
PDDL



Plan

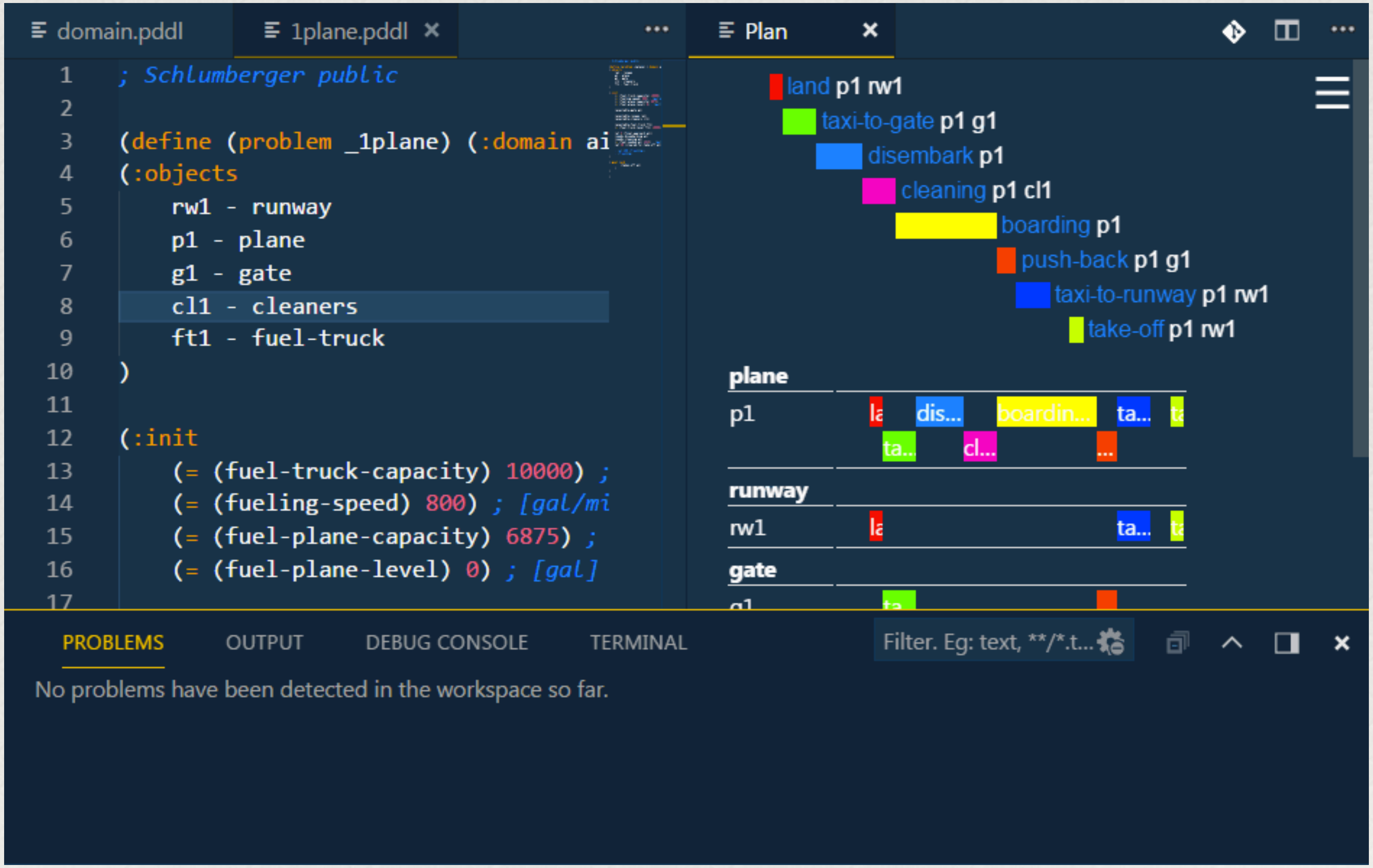
Sequence of actions

About a state-space planner (STRIPS)	
About actions, its admissibility and effects in the environment	About the domain, preferences and states



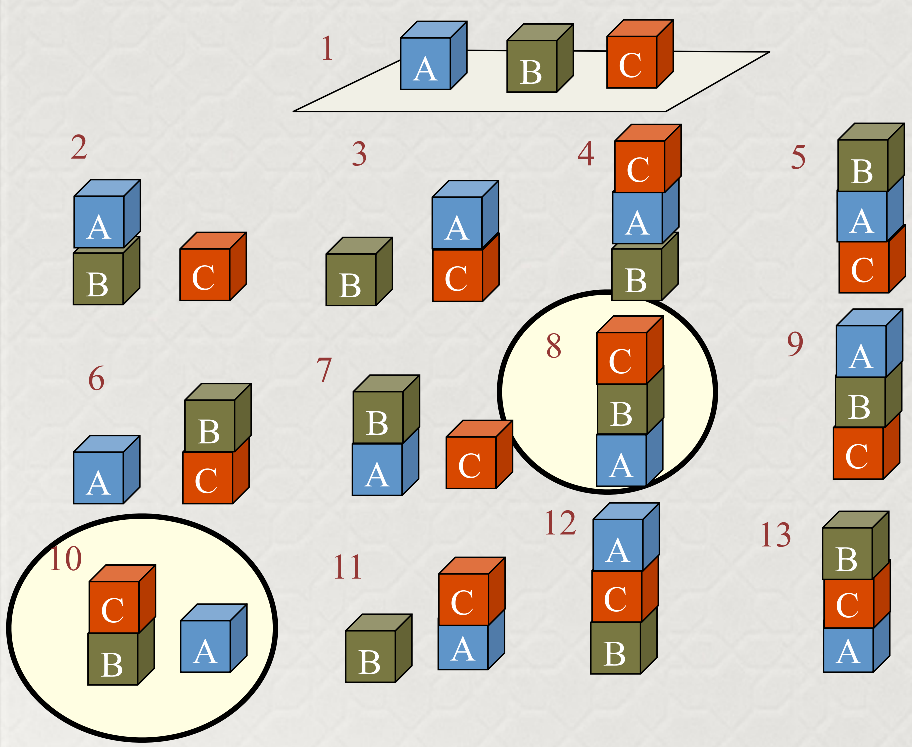
Automated planning and scheduling

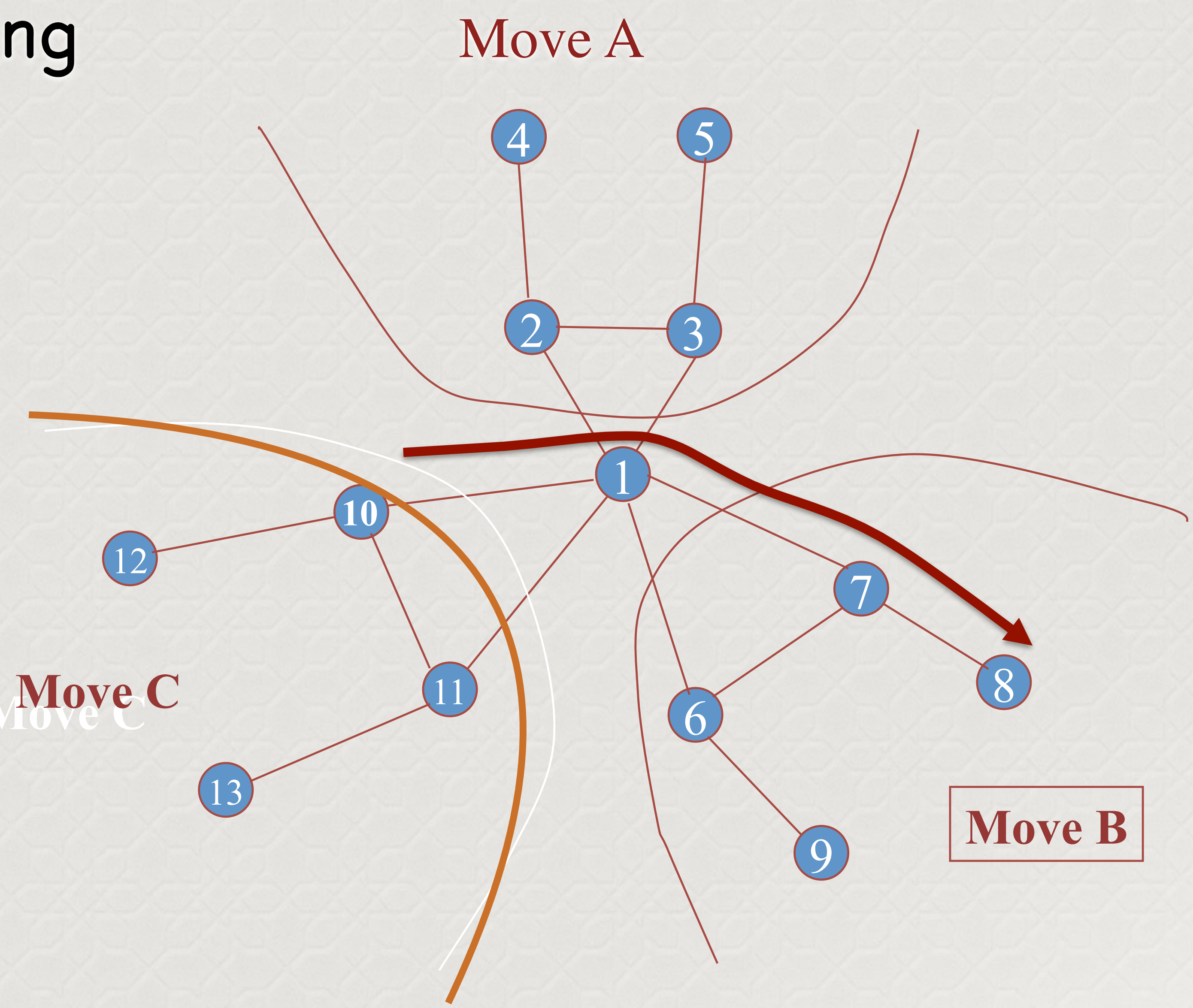
Problem Domain Definition Language



KEPS-Knowledge Engineering for Planning and Scheduling

About a state-space planner (STRIPS)	
About actions, its admissibility and effects in the environment	About the domain, preferences and states

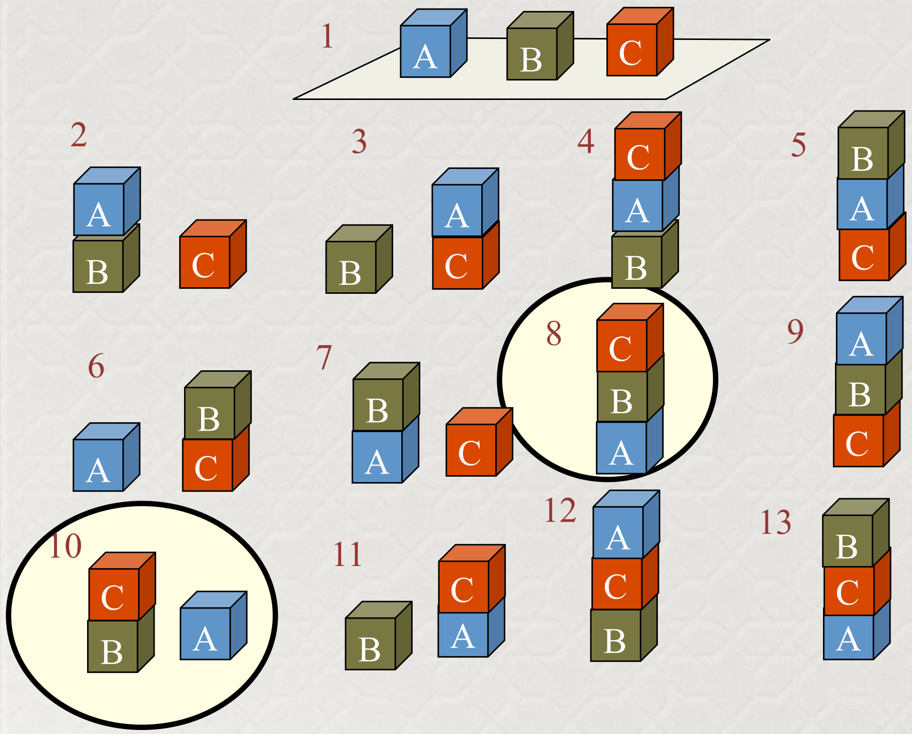


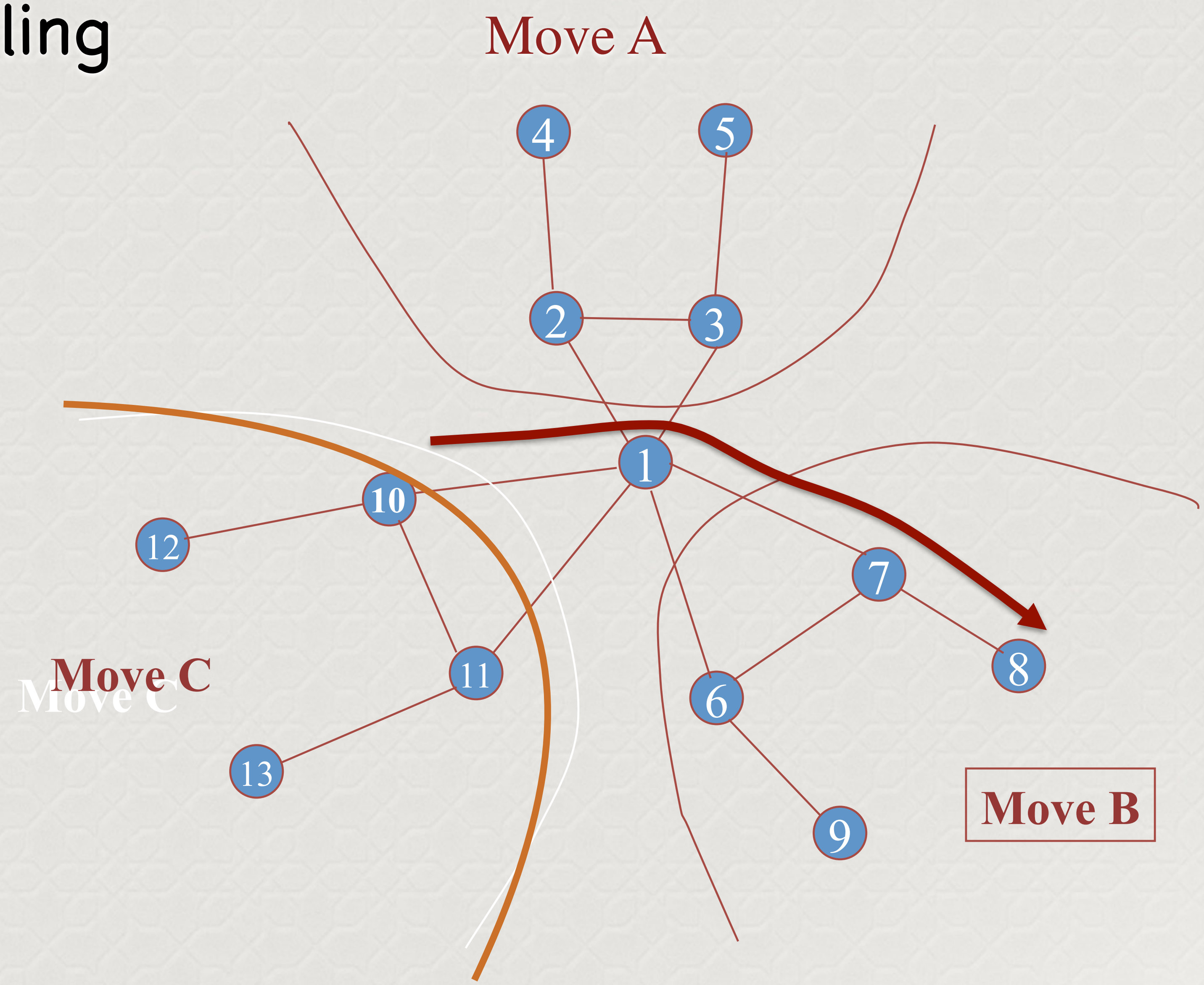


KEPS-Knowledge Engineering for Planning and Scheduling

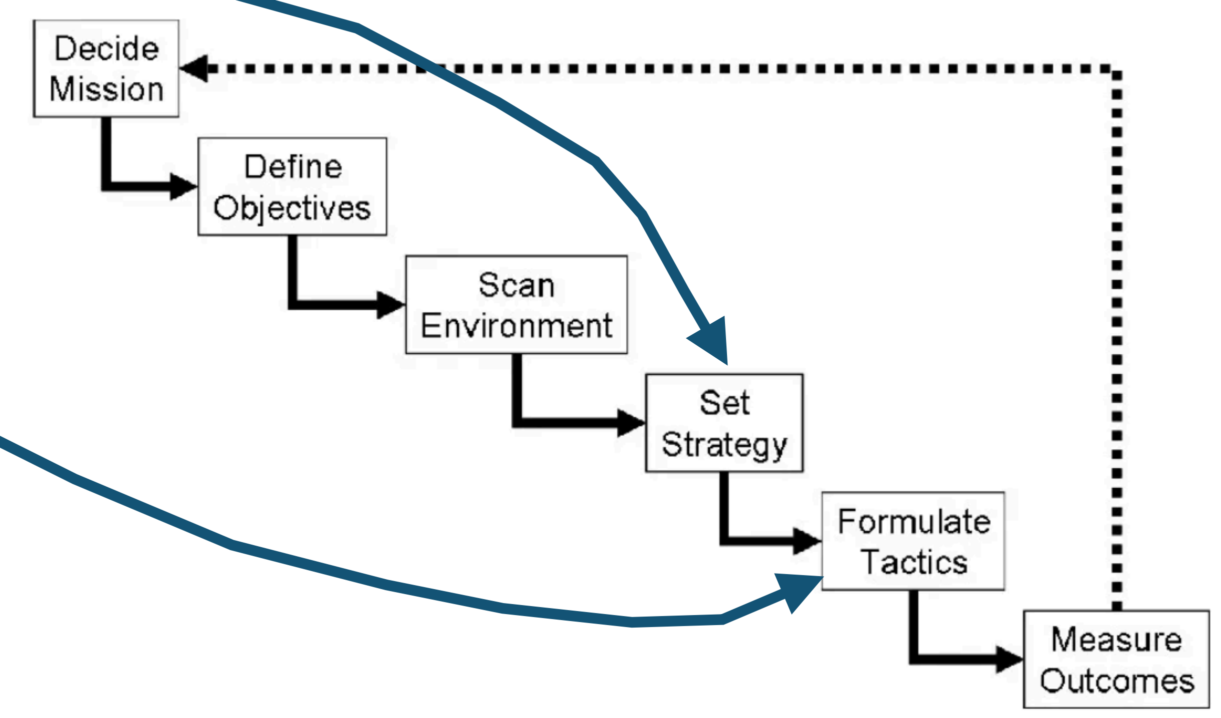
About a state-space planner (STRIPS)

About actions, its admissibility and effects in the environment	About the domain, preferences and states



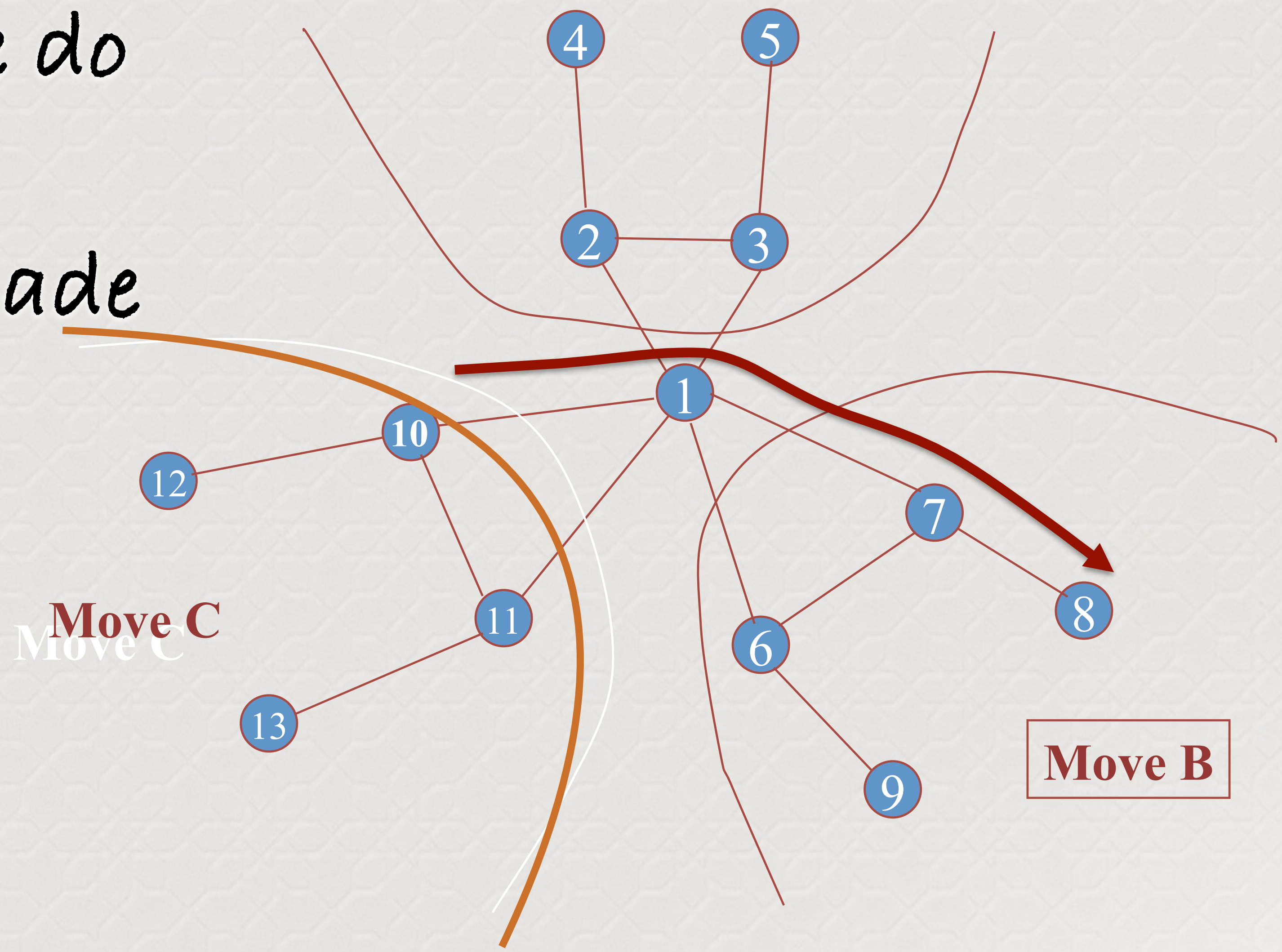


About a state-space planner (STRIPS)	
About actions, its admissibility and effects in the environment	About the domain, preferences and states



Automated planning and scheduling

Explorando a análise do domínio, podemos eliminar a possibilidade da Anomalia de Sussman?



Escola Politécnica da USP

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
2019, VOL. 57, NO. 15-16, 4766-4790
https://doi.org/10.1080/00207543.2018.1424372

Taylor & Francis
Taylor & Francis Group

Knowledge-based expert system in manufacturing planning: state-of-the-art review

S.P. Leo Kumar

Department of Production Engineering, PSG College of Technology, Coimbatore, India

ABSTRACT
In this paper, an effort has been made for intense review on Knowledge-Based Expert System (KB-ES) applications in manufacturing planning. Uniqueness of the present review work is addressed in terms of analysis on published review articles and their review gap. Research works exemplified between 1981 and 2016 were reviewed in terms of ES application in handling product variety, execution of process planning activities, machining, tool selection, tool design, welding, advanced manufacturing, product development. A statistical analysis was carried out in relation with number of publications, domain-specific area and their percentage contribution. It was inferred that, most of the work focused on ES applications related to tool design and machining apart from execution of various process planning activities. Future research can focus on the development frame-based, object oriented-based, ontology-based knowledge representation in order to develop robust system in decision-making for handling complex engineering problem. ES applications can be extended to field of micro fabrication, machine tool development and integrated system development from design to manufacturing.

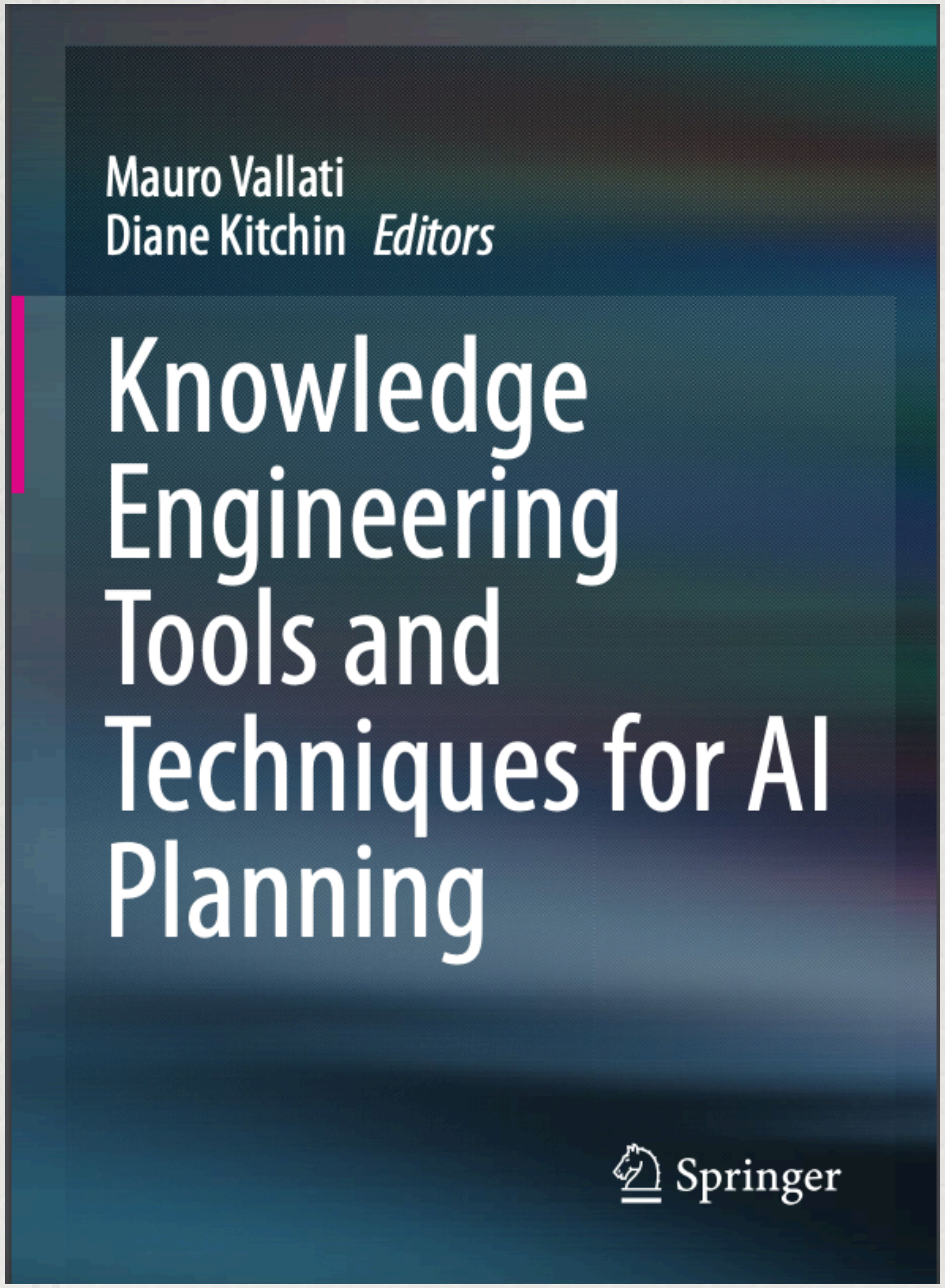
ARTICLE HISTORY
Received 17 July 2017
Accepted 30 December 2017

KEYWORDS
CAPP, artificial intelligence, manufacturing, knowledge-based system, expert system

CONTACT * Email: spleokumar@gmail.com

1. Introduction

Knowledge-based expert system or Expert System (ES) deals with computer programme that possess own decision-making capability to solve a problem of interest. ES concerned with creation of computational system that imitates intelligent behaviour of human expertise. John McCarthy invented



Chapter 3
Formal Knowledge Engineering for Planning: Pre and Post-Design Analysis

Jose Reinaldo Silva, Javier Martinez Silva, and Tiago Stegun Vaquero

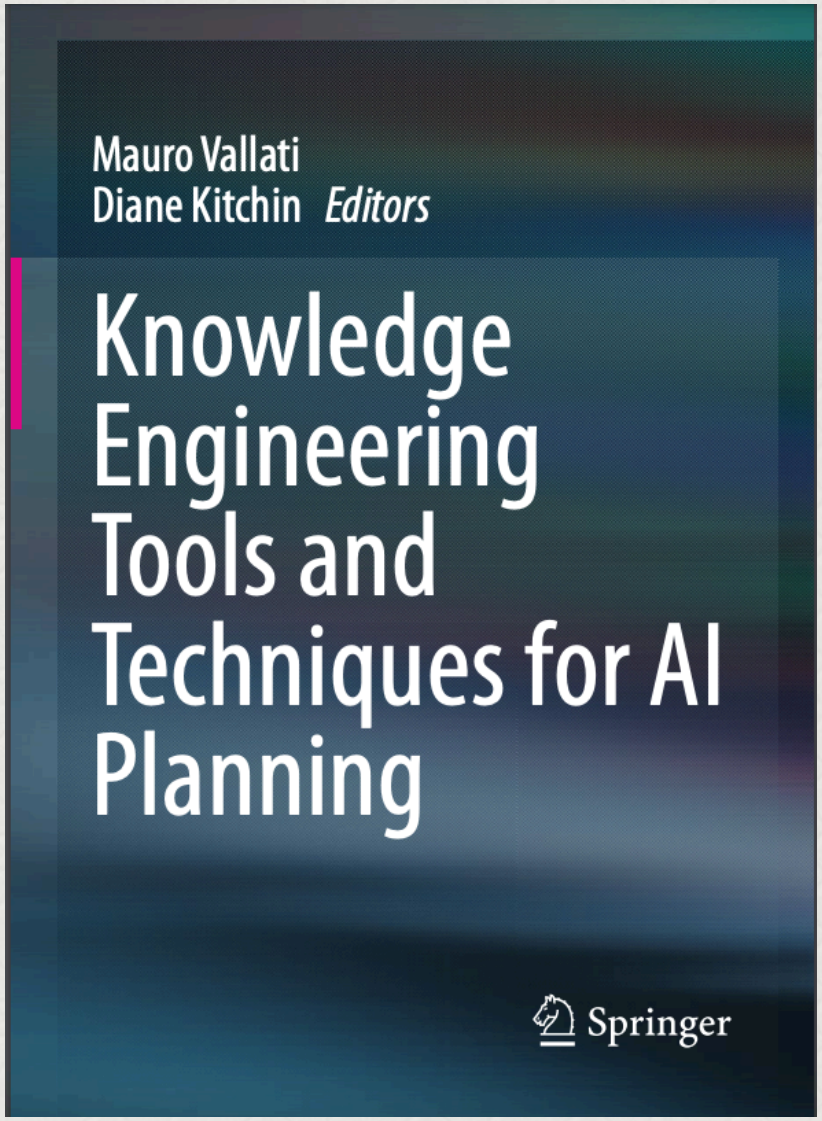
Abstract The interest and scope of the area of autonomous systems have been steadily growing in the last 20 years. Artificial intelligence planning and scheduling is a promising technology for enabling intelligent behavior in complex autonomous systems. To use planning technology, however, one has to create a knowledge base from which the input to the planner will be derived. This process requires advanced knowledge engineering tools, dedicated to the acquisition and formulation of the knowledge base, and its respective integration with planning algorithms that reason about the world to plan intelligently. In this chapter, we shortly review the existing knowledge engineering tools and methods that support the design of the problem and domain knowledge for AI planning and scheduling applications (AI P&S). We examine the state-of-the-art tools and methods of knowledge engineering for planning & scheduling (KEPS) in the context of an abstract design process for acquiring, formulating, and analyzing domain knowledge. Planning quality is associated with requirements knowledge (pre-design) which should match properties of plans (post-design). While examining the literature, we analyze the design phases that have not received much attention, and propose new approaches to that, based on theoretical analysis and also in practical experience in the implementation of the system itSIMPLE.

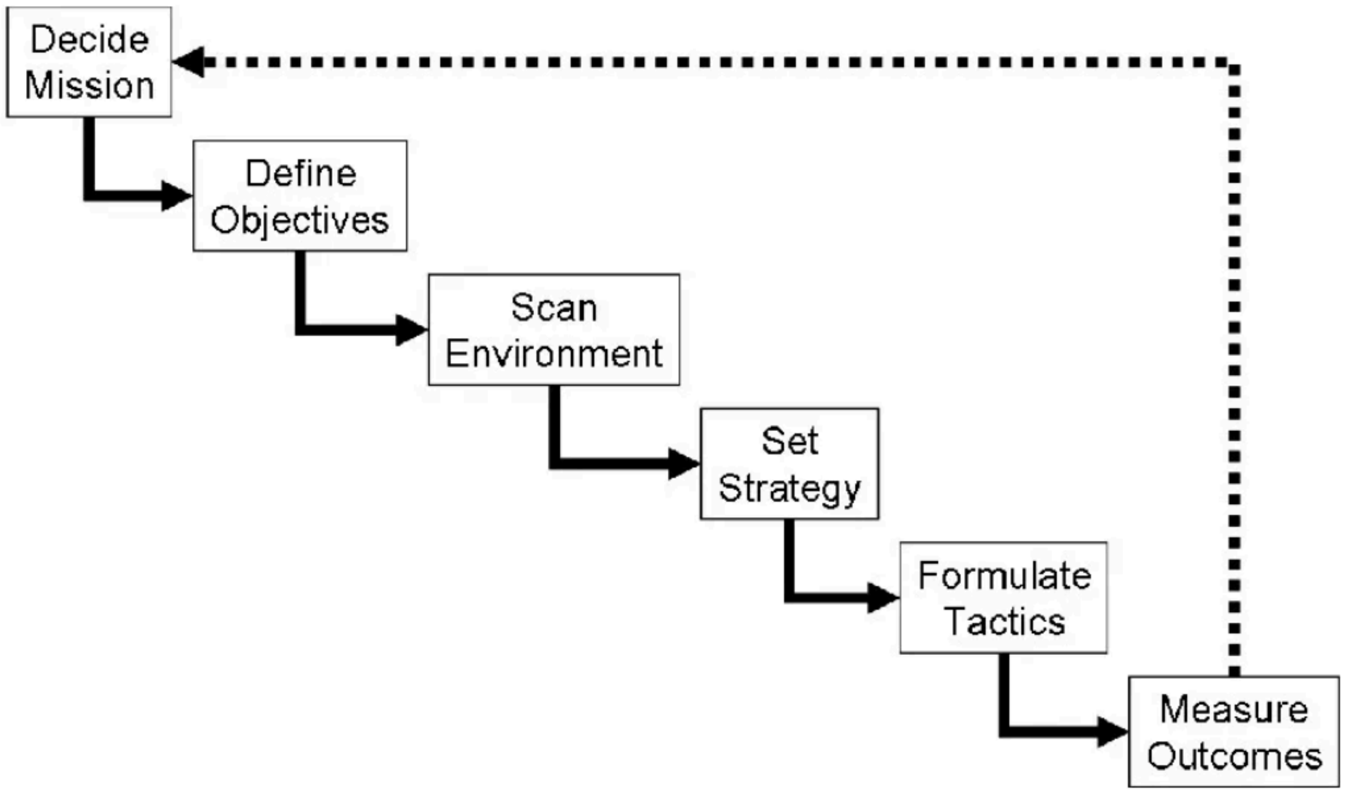
Keywords Planning design · Post-design analysis · Planning automation · Automation by planning

J. R. Silva (✉)
Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brazil
e-mail: reinaldo@usp.br; http://dlab.poli.usp.br

J. M. Silva

KEPS tools	Design Phases and Properties									
	Design Phases						Domain Independent	Planner Independent	Intended Expert	
	1	2	3	4	5	6			Domain	Planning
O-Plan		✓				≈	✓			✓
SIPE		✓					✓			✓
GIPO		✓	✓	✓	✓	✓	✓	≈	≈	✓
itSIMPLE	✓	✓	≈	✓	✓	✓	✓	✓	✓	✓
EUROPA		✓	✓		✓	✓	✓			✓
ModPlan		✓	≈		✓	✓	✓			✓
VIZ		✓		✓			✓		✓	
TIM			✓				✓			✓
DISCOPLAN			✓				✓	≈		✓
RSA			✓				✓			✓
RedOp			✓				✓	✓		✓
VAL						✓	✓	✓		✓
PlanWorks						✓	✓			✓
MrSPOCK						✓			✓	
JABBAH		✓		✓	✓	✓			✓	
PORSCE II				✓	✓			✓	✓	
CoastWatch						✓			✓	
FlowOpt		✓	✓		✓	✓			✓	
MARIO		✓		✓	✓				✓	
SLAF		✓					✓	✓		✓
LAMP		✓					✓	✓		✓
LOCM		✓					✓	≈		✓
ARMS		✓					✓	✓		✓
Bonasso & Boddy, 2010	✓			✓					✓	✓
Bouillet et al., 2007		✓					✓			✓
Fox & Long, 1999			✓				✓			✓
Crawford et al., 1996			✓				✓			✓
Fernández et al., 2009				✓				✓	✓	
Giuliano & Johnston, 2010						✓			✓	
Myers, 2006						✓	✓		✓	
Chrpa et al., 2012a						✓	✓	✓		✓
Chrpa et al., 2012b			✓				✓	✓		✓
Nakhost & Muller, 2010						✓	✓	✓		✓





Automated planning and scheduling

Fig. 3.2 Summary of available tools and methods in the Knowledge Engineering For Planning and Scheduling literature, Design phases: (1) Requirements, (2) Knowledge Modeling, (3) Model Analysis, (4) Model Preparation, (5) Plan/Schedule Synthesis, (6) Plan/Schedule Analysis and Post-Design. *Checkmark* means that the feature is present in the tool, *approx* means that it is to some degree present, and *blank* means that it is not present

Seja qual for a estratégia ou ciclo de projeto, é essencial separar a Engenharia de conhecimento do processo de resolução do "planning problem" - que seria o planejador.

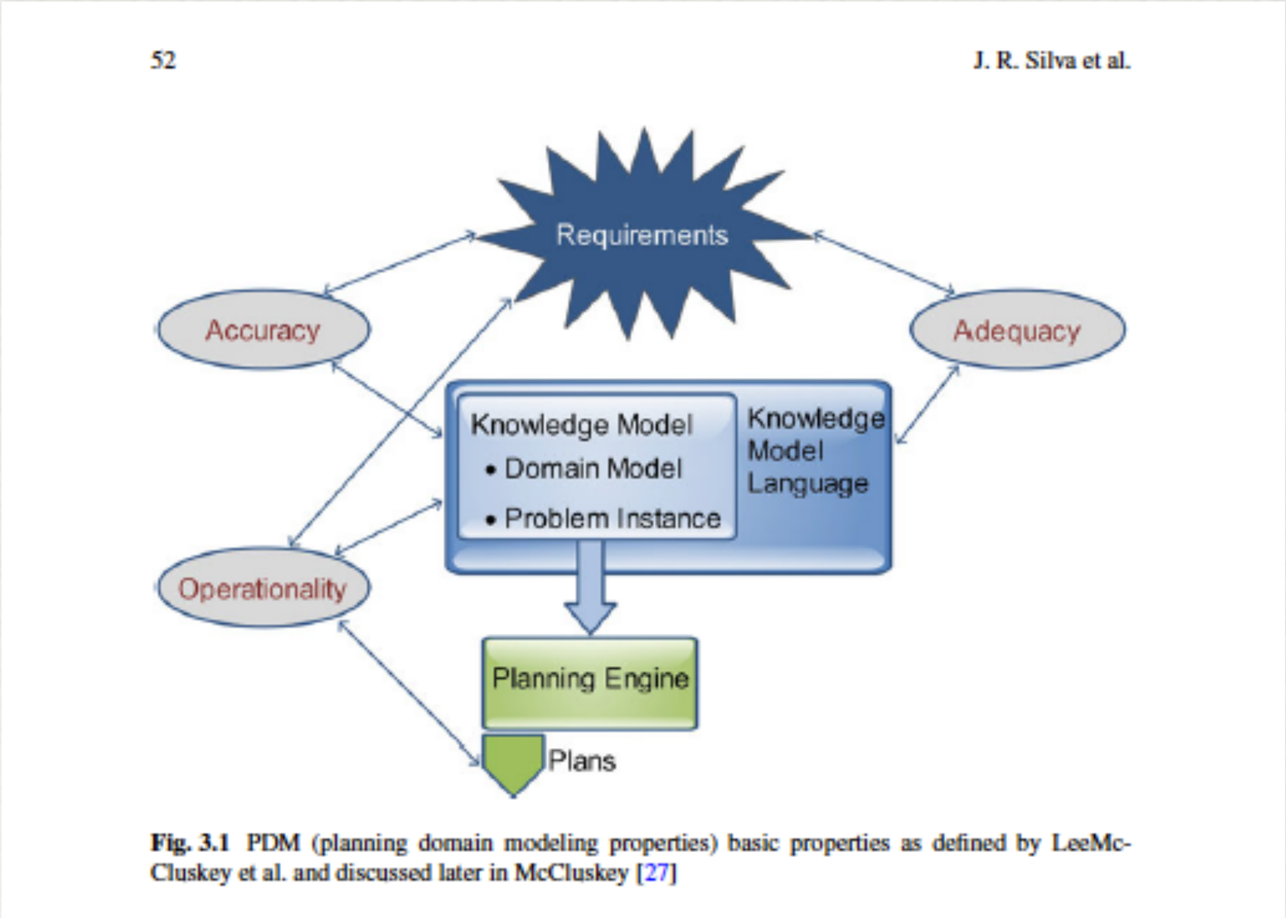
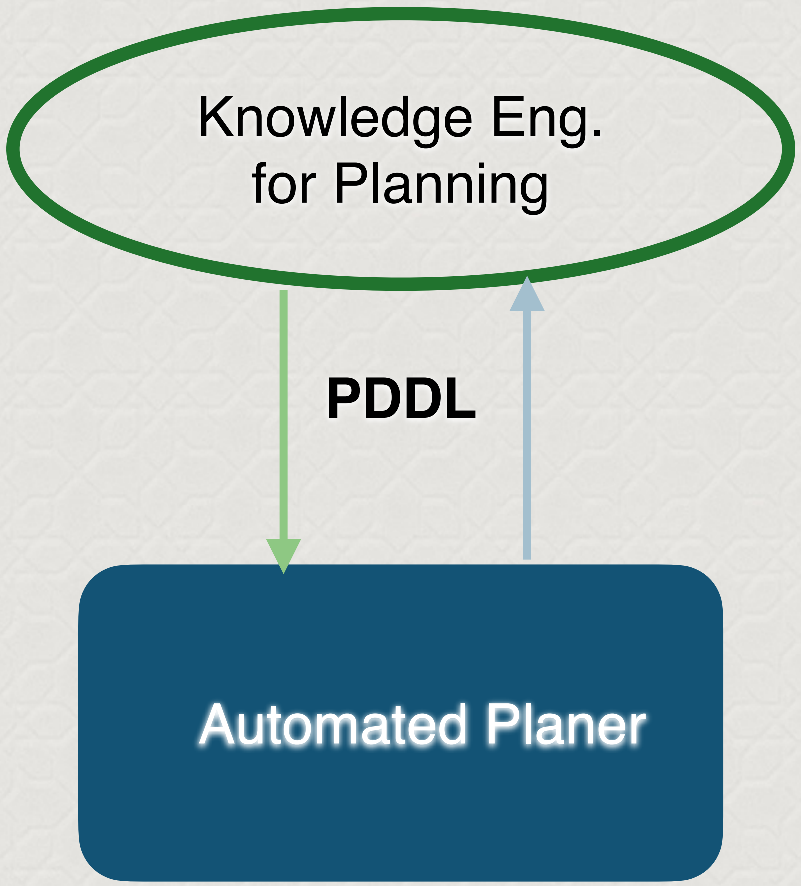

52 J. R. Silva et al.

Fig. 3.1 PDM (planning domain modeling properties) basic properties as defined by LeeMc-Cluskey et al. and discussed later in McCluskey [27]

Chamaremos "planning system" o sistema que inclui o sub-sistema de Enga. de Conhecimento e o planejador automático - outro sub-sistema.



A implicação lógica é que será necessário ter uma linguagem de transferência entre esses dois sub-sistemas.

Prof. Dr. Malte Helmert

What is PDDL?

PDDL = Planning Domain Definition Language

⇝ standard encoding language for “classical” planning tasks

Components of a PDDL planning task:

- **Objects:** Things in the world that interest us.
- **Predicates:** Properties of objects that we are interested in; can be *true* or *false*.
- **Initial state:** The state of the world that we start in.
- **Goal specification:** Things that we want to be true.
- **Actions/Operators:** Ways of changing the state of the world.

Prof. Dr. Malte Helmert

How to Put the Pieces Together

Planning tasks specified in PDDL are separated into two files:

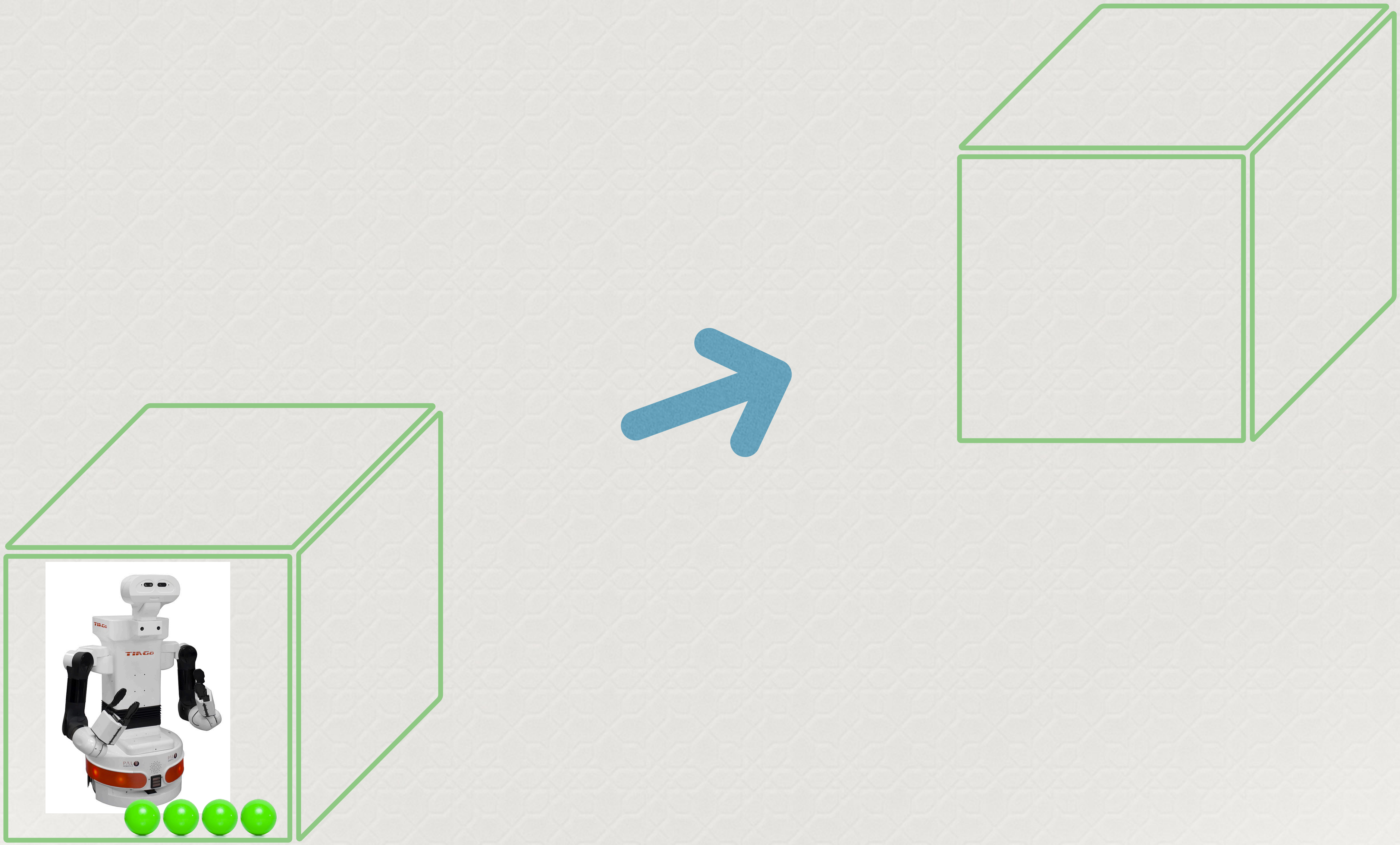
1. A **domain file** for predicates and actions.
2. A **problem file** for objects, initial state and goal specification.

Running Example

Gripper task with four balls:

There is a robot that can move between two rooms and pick up or drop balls with either of his two arms. Initially, all balls and the robot are in the first room. We want the balls to be in the second room.

- **Objects:** The two rooms, four balls and two robot arms.
- **Predicates:** Is x a room? Is x a ball? Is ball x inside room y? Is robot arm x empty? [...]
- **Initial state:** All balls and the robot are in the first room. All robot arms are empty. [...]
- **Goal specification:** All balls must be in the second room.
- **Actions/Operators:** The robot can move between rooms, pick up a ball or drop a ball.



Running Example

Gripper task with four balls:

There is a robot that can move between two rooms and pick up or drop balls with either of his two arms. Initially, all balls and the robot are in the first room. We want the balls to be in the second room.

- **Objects:** The two rooms, four balls and two robot arms.
- **Predicates:** Is x a room? Is x a ball? Is ball x inside room y? Is robot arm x empty? [...]
- **Initial state:** All balls and the robot are in the first room. All robot arms are empty. [...]
- **Goal specification:** All balls must be in the second room.
- **Actions/Operators:** The robot can move between rooms, pick up a ball or drop a ball.

Gripper task: Objects

Objects:

Rooms: `rooma`, `roomb`

Balls: `ball1`, `ball2`, `ball3`, `ball4`

Robot arms: `left`, `right`

In PDDL:

```
(:objects rooma roomb
          ball1 ball2 ball3 ball4
          left right)
```

Gripper task: Predicates

Predicates:

ROOM(x)	– true iff x is a room
BALL(x)	– true iff x is a ball
GRIPPER(x)	– true iff x is a gripper (robot arm)
at-robby(x)	– true iff x is a room and the robot is in x
at-ball(x, y)	– true iff x is a ball, y is a room, and x is in y
free(x)	– true iff x is a gripper and x does not hold a ball
carry(x, y)	– true iff x is a gripper, y is a ball, and x holds y

In PDDL:

```
(:predicates (ROOM ?x) (BALL ?x) (GRIPPER ?x)
             (at-robby ?x) (at-ball ?x ?y)
             (free ?x) (carry ?x ?y))
```

Gripper task: Initial state

Initial state:

ROOM(rooma) and ROOM(roomb) are true.
BALL(ball1), ..., BALL(ball4) are true.
GRIPPER(left), GRIPPER(right), free(left) and free(right) are true.
at-robby(rooma), at-ball(ball1, rooma), ..., at-ball(ball4, rooma) are true.
Everything else is false.

In PDDL:

```
(:init (ROOM rooma) (ROOM roomb)
       (BALL ball1) (BALL ball2) (BALL ball3) (BALL ball4)
       (GRIPPER left) (GRIPPER right) (free left) (free right)
       (at-robby rooma)
       (at-ball ball1 rooma) (at-ball ball2 rooma)
       (at-ball ball3 rooma) (at-ball ball4 rooma))
```

Gripper task: Goal specification

Goal specification:

`at-ball(ball1, roomb)`, ..., `at-ball(ball4, roomb)` must be true.
Everything else we don't care about.

In PDDL:

```
(:goal (and (at-ball ball1 roomb)
            (at-ball ball2 roomb)
            (at-ball ball3 roomb)
            (at-ball ball4 roomb)))
```

Gripper task: Movement operator

Action/Operator:

Description:	The robot can move from x to y.
Precondition:	ROOM(x), ROOM(y) and at-robby(x) are true.
Effect:	at-robby(y) becomes true. at-robby(x) becomes false. Everything else doesn't change.

In PDDL:

```
(:action move :parameters (?x ?y)
   :precondition (and (ROOM ?x) (ROOM ?y)
                      (at-robby ?x))
   :effect       (and (at-robby ?y)
                      (not (at-robby ?x))))
```

Gripper task: Pick-up operator

Action/Operator:

Description: The robot can pick up x in y with z.

Precondition: BALL(x), ROOM(y), GRIPPER(z), at-ball(x, y), at-robby(y) and free(z) are true.

Effect: carry(z, x) becomes true. at-ball(x, y) and free(z) become false. Everything else doesn't change.

In PDDL:

```
(:action pick-up :parameters (?x ?y ?z)
   :precondition (and (BALL ?x) (ROOM ?y) (GRIPPER ?z)
                      (at-ball ?x ?y) (at-robby ?y) (free ?z))
   :effect       (and (carry ?z ?x)
                      (not (at-ball ?x ?y)) (not (free ?z))))
```

Gripper task: Drop operator

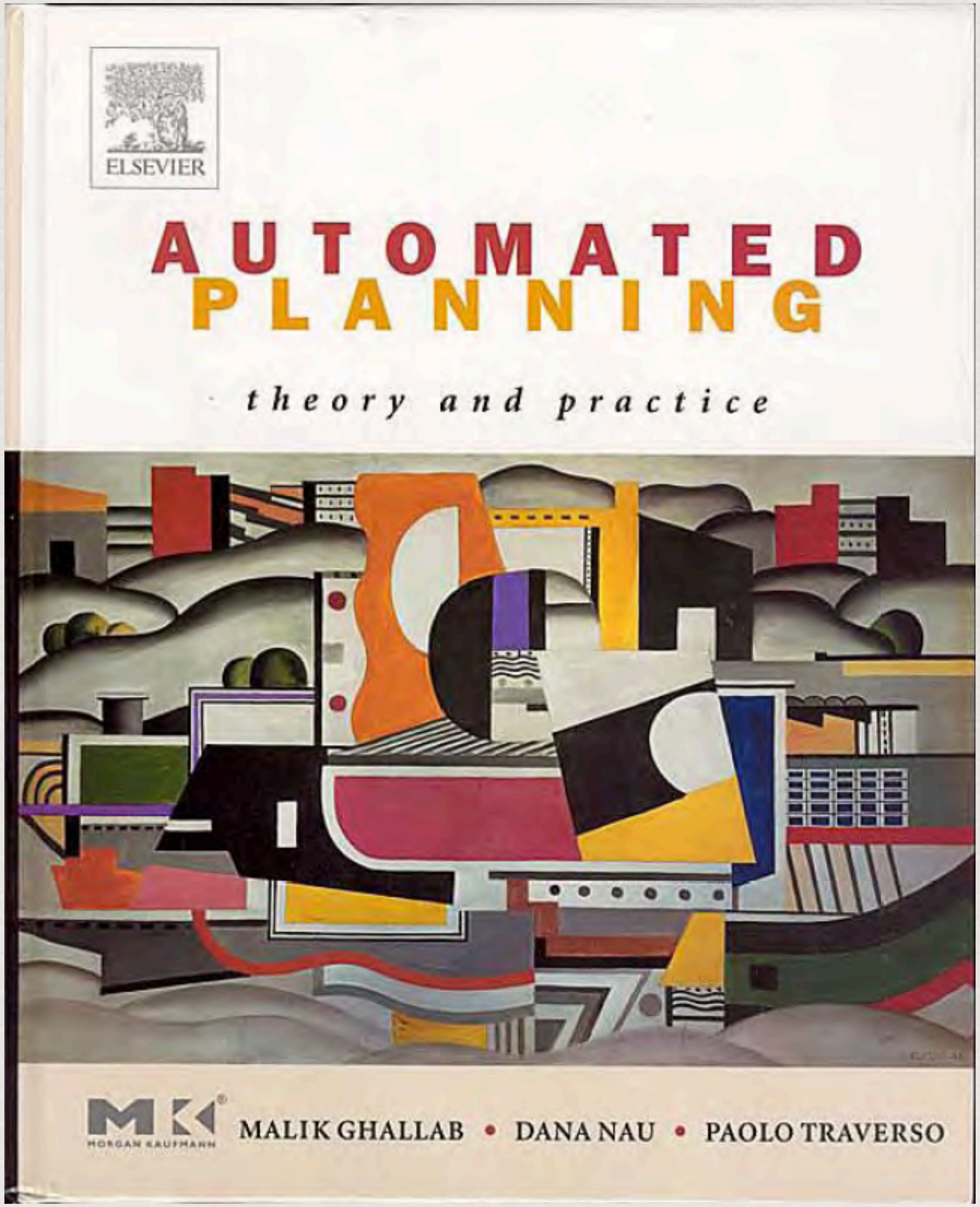
Action/Operator:

Description: The robot can drop x in y from z.

(Preconditions and effects similar to the pick-up operator.)

In PDDL:

```
(:action drop :parameters (?x ?y ?z)
   :precondition (and (BALL ?x) (ROOM ?y) (GRIPPER ?z)
                      (carry ?z ?x) (at-robby ?y))
   :effect       (and (at-ball ?x ?y) (free ?z)
                      (not (carry ?z ?x))))
```


ELSEVIER
AUTOMATED
PLANNING
theory and practice
MALIK GHALLAB • DANA NAU • PAOLO TRAVERSO
MORGAN KAUFMANN

planning.domains

PDDL Editor | File | Session | Import | Solve | Plugins | Help | planning.domains

unnamed1.pddl

```
Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(P1, SFO) ∧ At(P2, JFK)
            ∧ Cargo(C1) ∧ Cargo(C2) ∧ Plane(P1) ∧ Plane(P2)
            ∧ Airport(JFK) ∧ Airport(SFO))

Goal(At(C1, JFK) ∧ At(C2, SFO))

Action(Load(c, p, a),
       PRECOND: At(c, a) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
       EFFECT: ¬ At(c, a) ∧ In(c, p))

Action(Unload(c, p, a),
       PRECOND: In(c, p) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
       EFFECT: At(c, a) ∧ ¬ In(c, p))

Action(Fly(p, from, to),
       PRECOND: At(p, from) ∧ Plane(p) Airport(from) ∧ Airport(to)
       EFFECT: ¬ At(p, from) ∧ At(p, to))
```

Init(At(C1, SFO) $\wedge$ At(C2, JFK) $\wedge$ At(P1, SFO) $\wedge$ At(P2, JFK)
$\wedge$ Cargo(C1) $\wedge$ Cargo(C2) $\wedge$ Plane(P1) $\wedge$ Plane(P2)
$\wedge$ Airport(JFK) $\wedge$ Airport(SFO))

Goal(At(C1, JFK) $\wedge$ At(C2, SFO))

Action(Load(c, p, a),
PRECOND: At(c, a) $\wedge$ At(p, a) $\wedge$ Cargo(c) $\wedge$ Plane(p) $\wedge$ Airport(a)
EFFECT: $\neg$ At(c, a) $\wedge$ In(c, p))

Action(Unload(c, p, a),
PRECOND: In(c, p) $\wedge$ At(p, a) $\wedge$ Cargo(c) $\wedge$ Plane(p) $\wedge$ Airport(a)
EFFECT: At(c, a) $\wedge$ $\neg$ In(c, p))

Action(Fly(p, from, to),
PRECOND: At(p, from) $\wedge$ Plane(p) Airport(from) $\wedge$ Airport(to)
EFFECT: $\neg$ At(p, from) $\wedge$ At(p, to))

Entre os métodos clássicos da IA os sistemas baseados em conhecimento e os sistemas para planning & scheduling são os mais destacados. É possível aumentar significativamente a complexidade na direção dos deep knowledge systems, capazes de analisar e corrigir seus próprios planos.

O nosso curso PMR3510 termina aqui (exceto pela avaliação dos exercícios).
Desejo a todos boa sorte na carreira que começa!

Obrigado!