Model-Based Explainable AI for Safe and Trusted Human-Autonomy Teaming

Daniele Magazzeni

KING'S College LONDON

@DanMagazzeni

Trusted Autonomous Systems Hub

To facilitate co-creation with industrial partners, patents, spin out, joint big grant proposals, engagement with general audience.

Artificial Intelligence Planning

5G and Internet of Skills

Software Engineering

Verification

Argumentation

Provenance

Cyber Security

Social Science

Law School

Business School

Digital Humanities

Policy Institute

AI

Data
Driven

AI

Data Driven

Model Based

Because:

you don't have data to learn from
you don't have time to learn
your model evolves/changes

AI

Descriptive

Data Driven

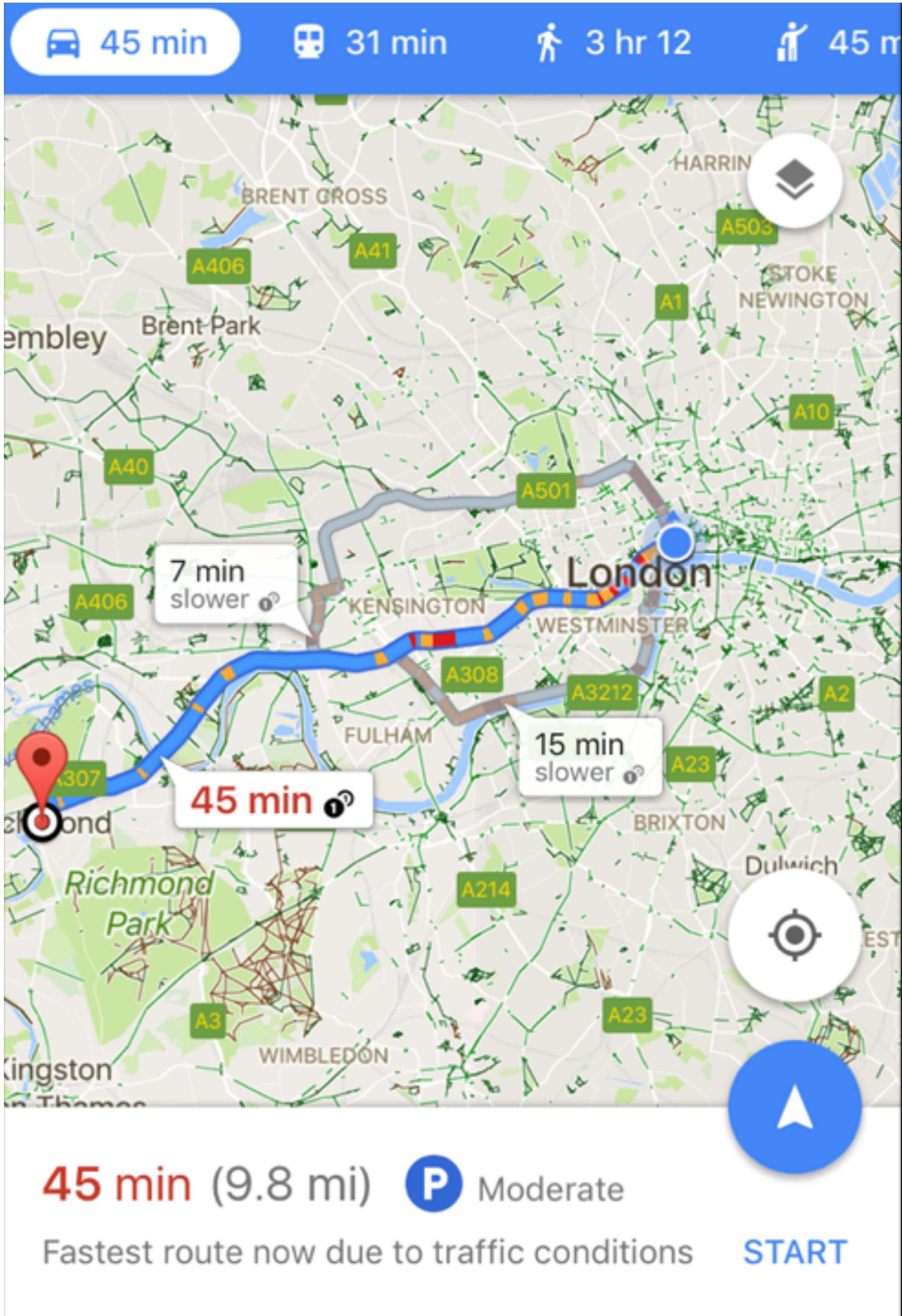
Model Based

Learned-Model Based AI

e:

ou don’t have data to learn from
you don’t have time to learn
your model evolves/changes

you care about safety and trust



AI Planning

INPUT:

-Model of your domain
-Initial condition (sensor data)
-Goal

OUTPUT:

-Plan (metric optimisation)

Plans are found in (milli)seconds

With the same model you can set different goals

Artificial Intelligence Planning at King’s

- We create ***Planners*** to **assist humans** and for **autonomy**.
- A planner uses a model of an application domain and a description of a specific problem (starting point and goals) and generates a plan.
- If something changes, or need to achieve a new goal, just replan!
- **Planning** is combined with **Machine Learning** for demand prediction and policy generation
- We have a very rich portfolio of planning for **real applications**, with companies and organisations:
  - –Autonomous **Underwater** Vehicles
  - –Autonomous **Drones** and UAVs
  - –Multiple **Battery** System Management
  - –Air Traffic Control and Plane Taxiing
  - –**Logistics**
  - -**Energy** Technology
  - -Ocean Liners
  - -**Hybrid Vehicles**
  - -Urban Traffic Control
  - -Satellites


Model
Data
PLANNER
Plan

PDDL: Planning Domain Definition Language

```
(:durative-action do_hover
 :parameters (?v - vehicle ?from ?to - waypoint)
 :duration ( = ?duration (* (distance ?from ?to)
                            (invtime ?v)))
 :condition (and (at start (at ?v ?from))
                 (at start (connected ?from ?to)))
 :effect (and (at start (not (at ?v ?from)))
              (at end (at ?v ?to))))

(:durative-action observe
 :parameters (?v - vehicle ?wp - waypoint
              ?ip - inspectionpoint)
 :duration ( = ?duration (obstime))
 :condition (and (at start (at ?v ?wp))
                 (at start (cansee ?v ?ip ?wp)))
 :effect (and (at start (not (cansee ?v ?ip ?wp)))
              (at end (increase (observed ?ip)
                      (obs ?ip ?wp)))))
```

Temporal planning with time windows

```
(:durative-action do_hover_controlled ...)

(:durative-action do_hover_fast ...)

(:durative-action correct_position ...)

(:durative-action observe_inspection_point ...)

(:durative-action illuminate_pillar ...)

(:durative-action observe_pillar ...)

(:durative-action examine_panel ...)

(:durative-action turn_valve ...)

(:durative-action recalibrate_arm ...)
```

```
;; time window 2 [400--800]
(at 400 (= (valve_goal v2 270))
(at 400 (not (valve_blocked v2)))
(at 400 (valve_free v2))
(at 400 (not (valve_goal_unchecked v2)))

(at 800 (valve_blocked v2))
(at 800 (not (valve_free v2)))

(at 400 (= (valve_goal v3) 10))
(at 400 (not (valve_blocked v3)))
(at 400 (valve_free v3))
(at 400 (not (valve_goal_unchecked v3)))
```

```
0.000: (correct_position auv wp0)  [10.000]
10.001: (do_hover_controlled auv wp0 wp_strat_p0)  [33.532]
43.534: (turn_valve auv wp_strat_p0 p0 v0)  [120.000]
163.535: (correct_position auv wp_strat_p0)  [10.000]
173.536: (turn_valve auv wp_strat_p0 p0 v1)  [120.000]
293.537: (correct_position auv wp_strat_p0)  [10.000]
293.537: (recalibrate_arm auv wp0)  [180.000]
473.538: (turn_valve auv wp_strat_p0 p0 v2)  [120.000]
593.539: (correct_position auv wp_strat_p0)  [10.000]
603.540: (turn_valve auv wp_strat_p0 p0 v3)  [120.000]
```

PDDL: Planning Domain Definition Language

Planners are Domain-Independent
They are based on heuristic search

KCL Planners

Linear dynamics: **POPF/Optic/Colin**

-Forward heuristic search

-Use Linear Programming and Simple Temporal Networks to check temporal constraints

Polynomial Non-Linear dynamics: **SMTPlan**

-Encode the planning problem as SMT formula

-Use Computer Algebra System to compute indefinite integrals

Non-Linear dynamics: **UPMurphi/DiNO**

-Forward heuristic search

-Use discretisation to handle complex dynamics

All planners are open source

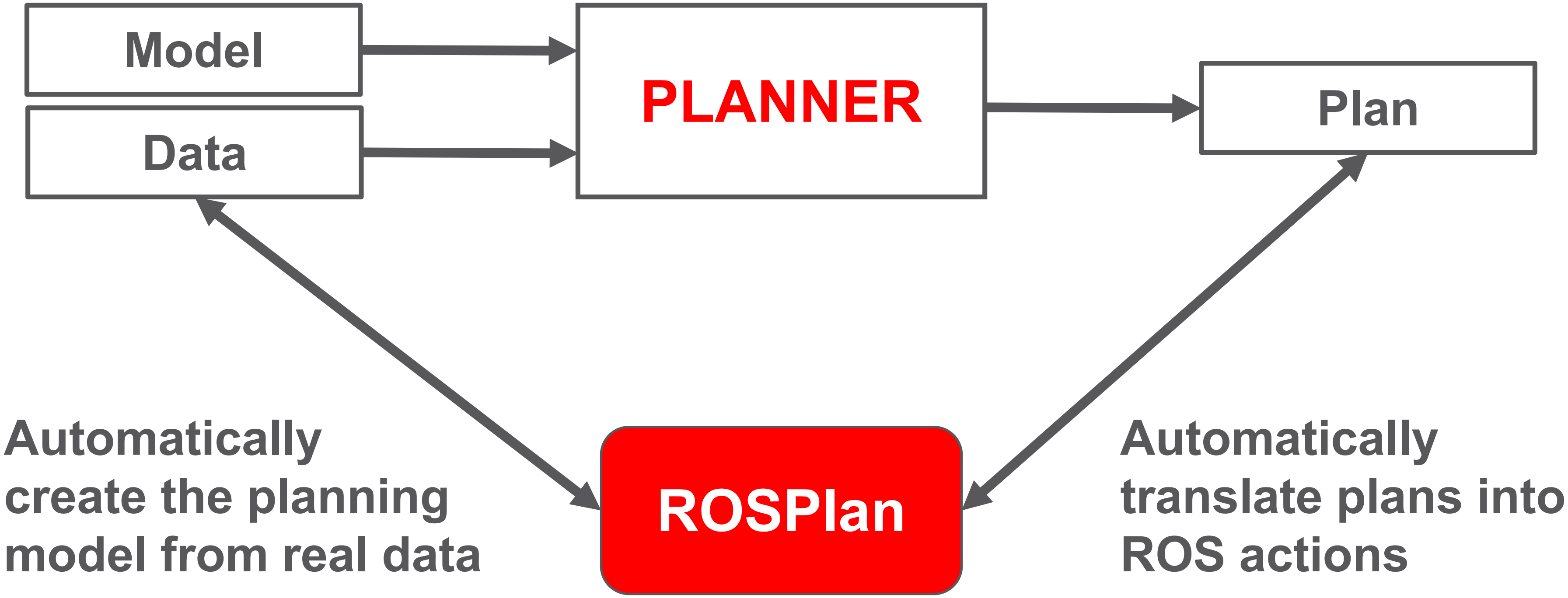
If you want to use AI for real…

…there are some key issues:

-Reality is always different from what you modelled (Replanning)
-Real-world is full of uncertainty
-Creating a plan is difficult, executing a plan is very difficult
-Real problems have huge state space
-"Task allocation" is only one (small) part of the problem
-Trust and Confidence
-Human-Autonomy Teaming

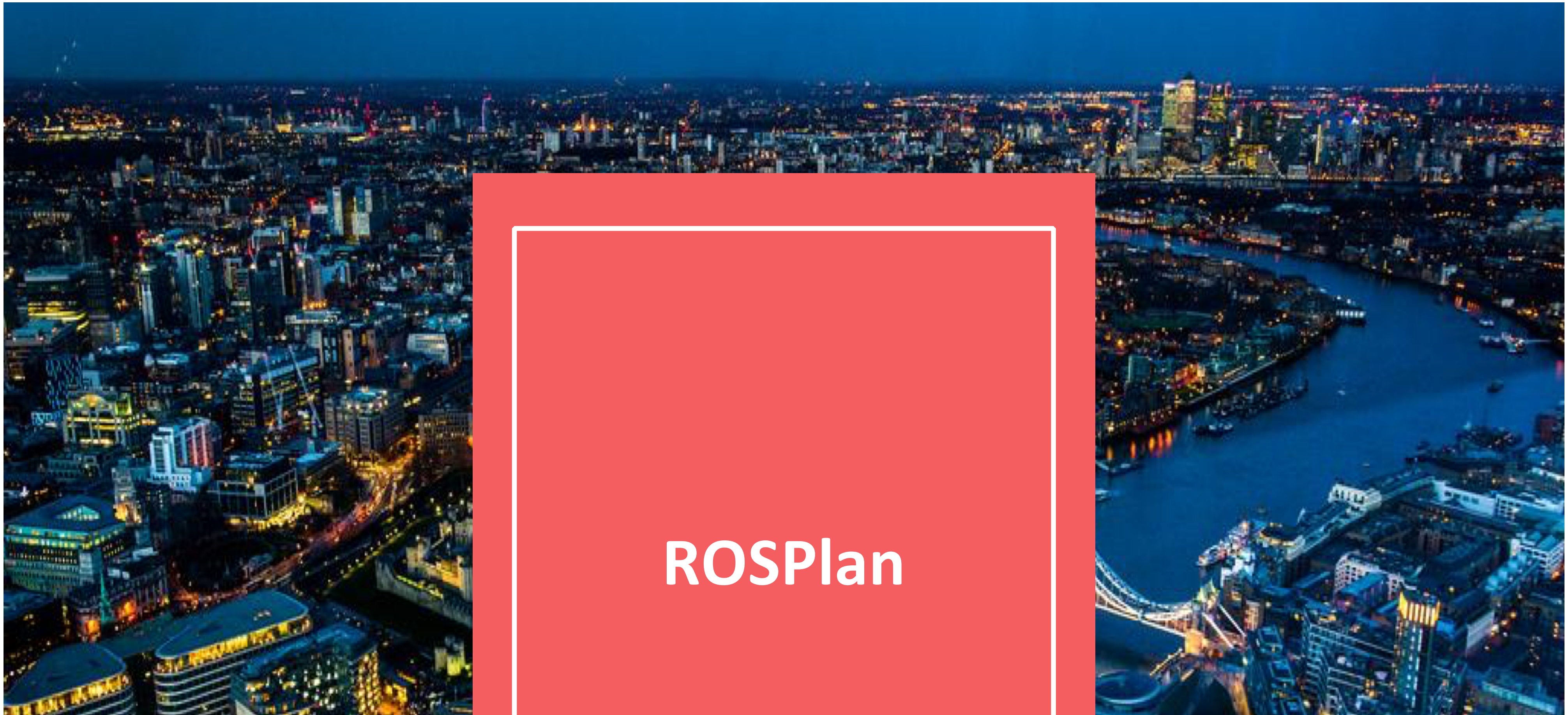

Model
Data
PLANNER
Plan

ROSPlan



Plan execution
Replanning
Plan failures
Model changes (e.g. equipment failures)
Probabilistic Planning

KING'S College LONDON



ROSPlan

Special thanks to
Dr Michael Cashmore

What is ROSPlan?

Tools for AI Planning in a ROS system.

ROSPlan has a set of default nodes which encapsulate **model revision, planning,** and **plan execution.** It allows a ROS system to produce and execute PDDL2.1 plans.

ROSPlan has a modular design, intended to be modified. It serves as a framework to test new modules with minimal effort.

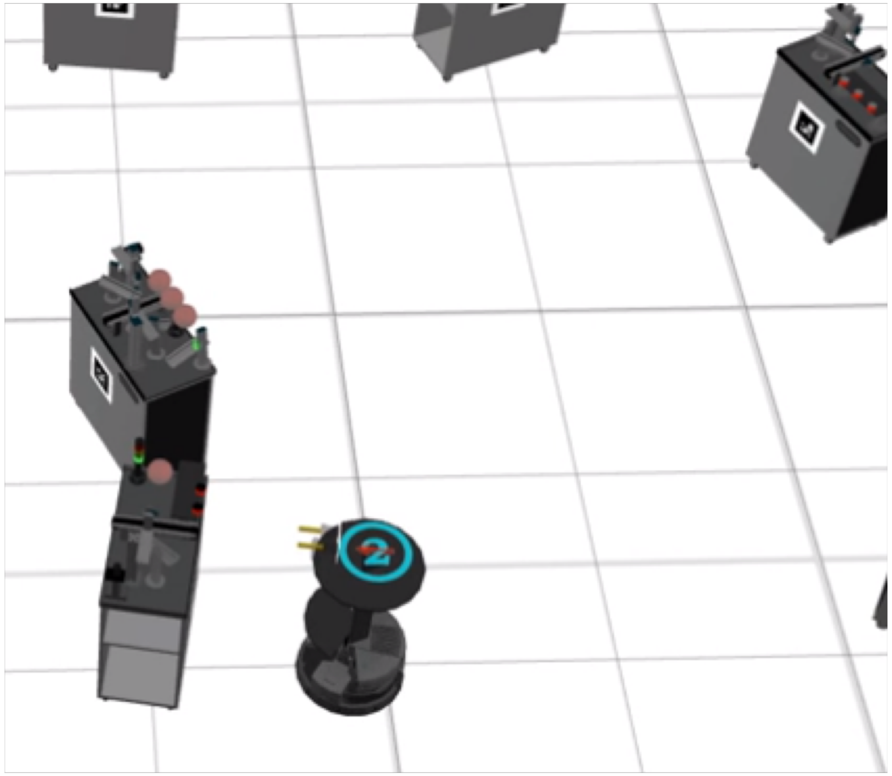
Komodo

Girona 500 AUV

SQUIRREL

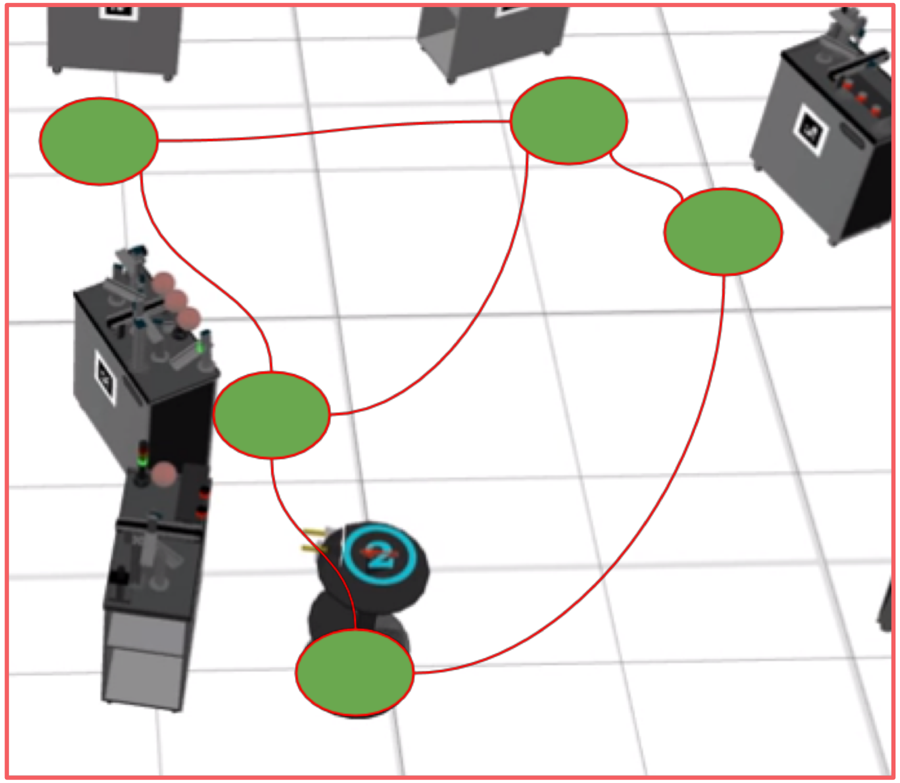
Nessie AUV

Model Revision



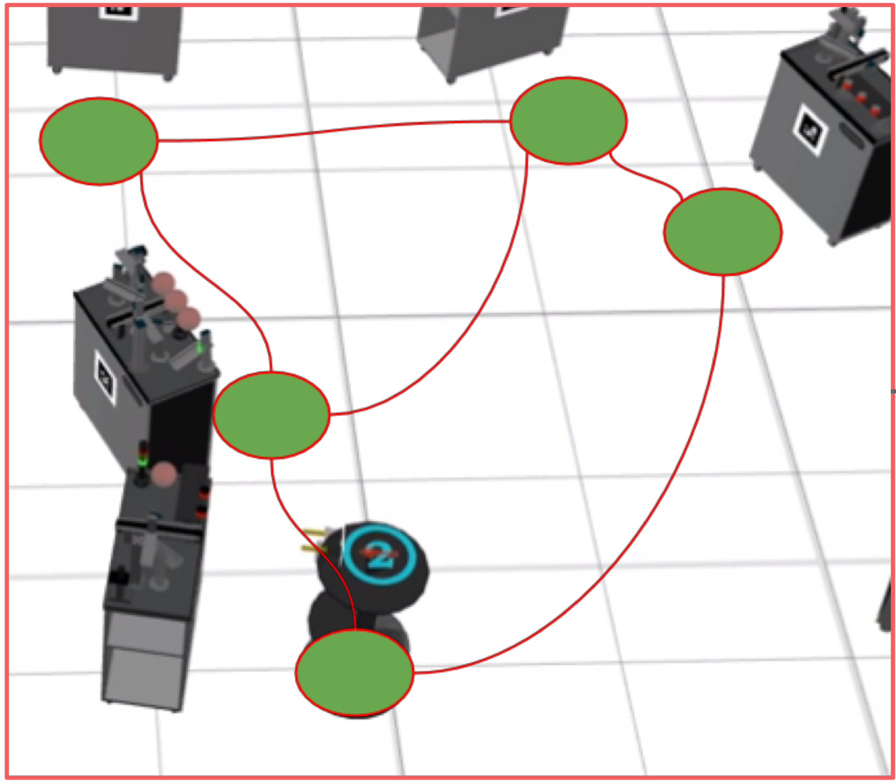
Sensor data is **continuously** parsed to form a symbolic representation of the **current** state.

Model Revision



Sensor data is **continuously** parsed to form a symbolic representation of the **current** state.

Model Revision



Sensor data is **continuously** parsed to form a symbolic representation of the **current** state.

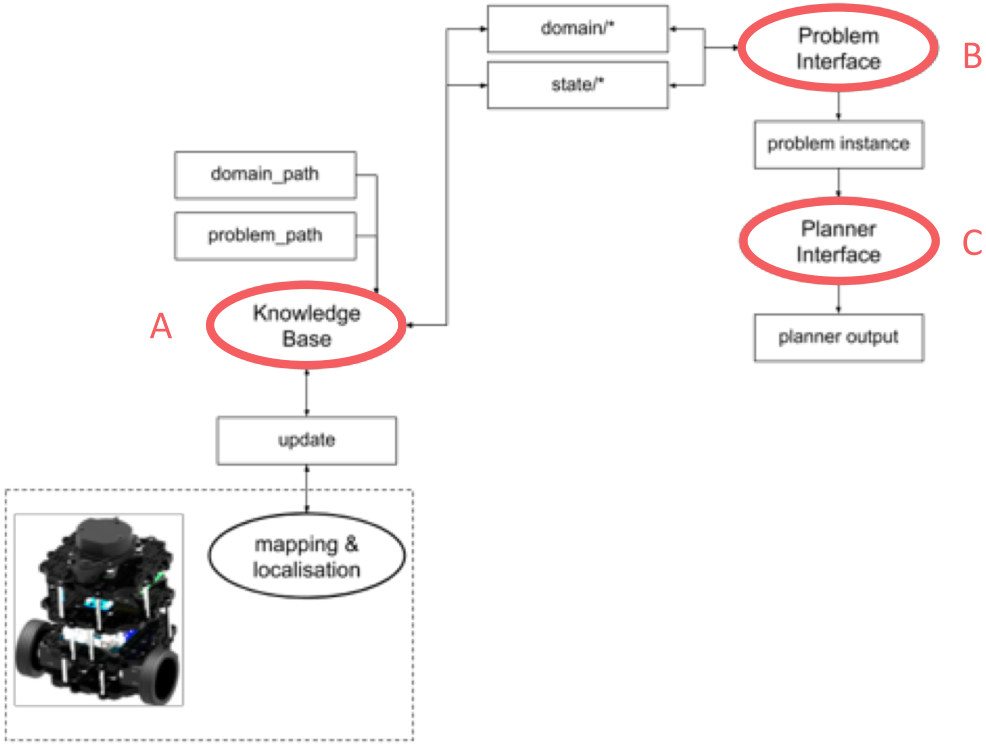
The state description is automatically converted into a model in PDDL 2.1 syntax.

```
(define (problem task)
(:objects
   wp0 wp1 wp2 wp3 wp4 wp5 - waypoint
   kenny - robot
)
(:init
   (robot_at kenny wp0)     (connected wp0 wp2)
   (connected wp0 wp4)      (connected wp1 wp0)
   (connected wp1 wp2)      ...
```

Model Revision

ROSPlan provides default nodes for:

A. Storing a representation of the state, with services for continuous update.
B. Producing problems in PDDL 2.1 syntax.
C. Passing the problems to the AI task planner to produce a plan.

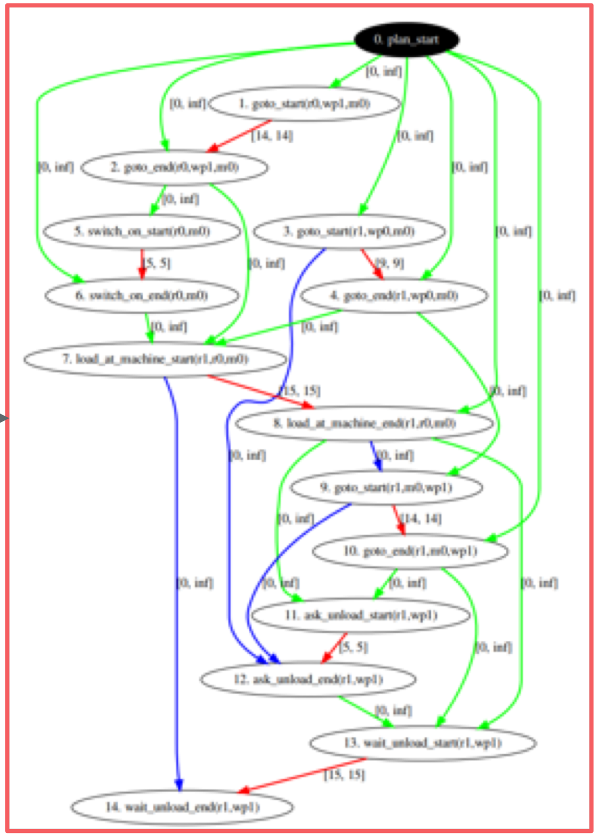


How Plans are Dispatched

```
0.000:     (goto r0 wp1 m0)                [14.000]
0.000:     (goto r1 wp0 m0)                [ 9.000]
14.001:    (switch_on r0 m0                [ 5.000]
19.002:    (load_at_machine r1 r0 m0) [15.000]
34.002:    (goto r1 m0 wp1)                [14.000]
48.002:    (ask_unload r1 wp1)             [ 5.000]
53.003:    (wait_unload r1 wp1)            [15.000]
```

How Plans are Dispatched

```
0.000:    (goto r0 wp1 m0)                [14.000]
0.000:    (goto r1 wp0 m0)                [ 9.000]
14.001:   (switch_on r0 m0                [ 5.000]
19.002:   (load_at_machine r1 r0 m0)      [15.000]
34.002:   (goto r1 m0 wp1)                [14.000]
48.002:   (ask_unload r1 wp1)             [ 5.000]
53.003:   (wait_unload r1 wp1)            [15.000]
```

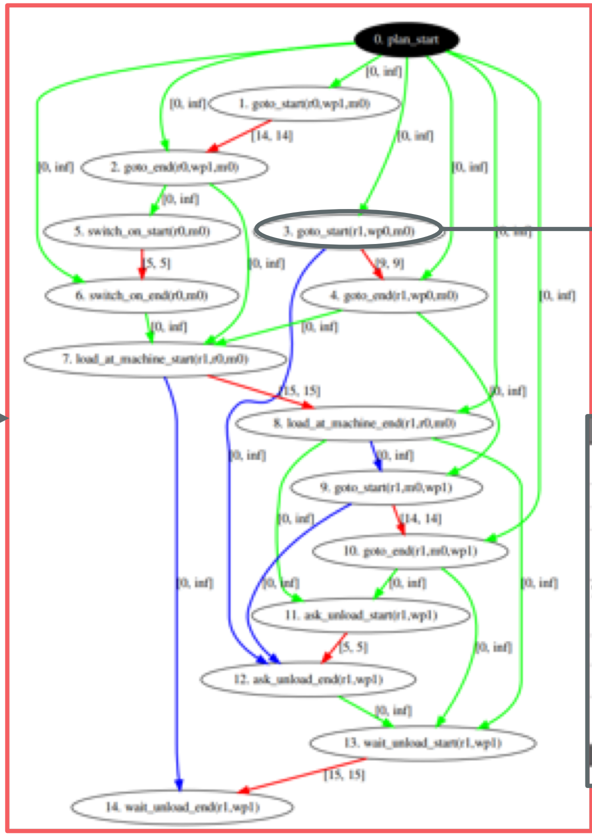


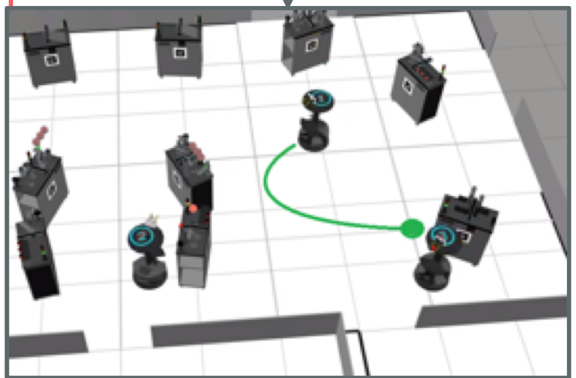
The PDDL plan is processed into an executable form.

How Plans are Dispatched

Actions in the plan will be dispatched at correct times as goals for lower-level control.

0.000:	(goto r0 wp1 m0)	[14.000]
0.000:	(goto r1 wp0 m0)	[9.000]
14.001:	(switch_on r0 m0	[5.000]
19.002:	(load_at_machine r1 r0 m0)	[15.000]
34.002:	(goto r1 m0 wp1)	[14.000]
48.002:	(ask_unload r1 wp1)	[5.000]
53.003:	(wait_unload r1 wp1)	[15.000]



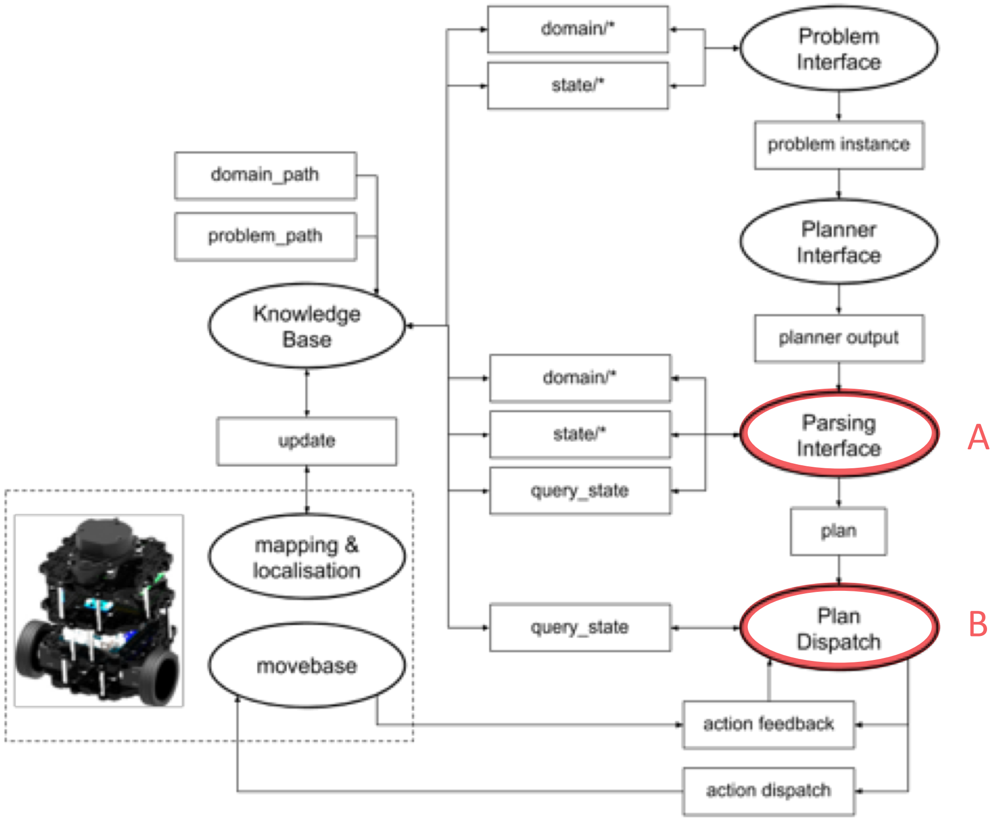


The PDDL plan is processed into an executable form.

How Plans are Dispatched

ROSPlan provides the nodes for:

A. Post-processing the plan to an executable form.
B. Executing the plan and dispatching the actions.



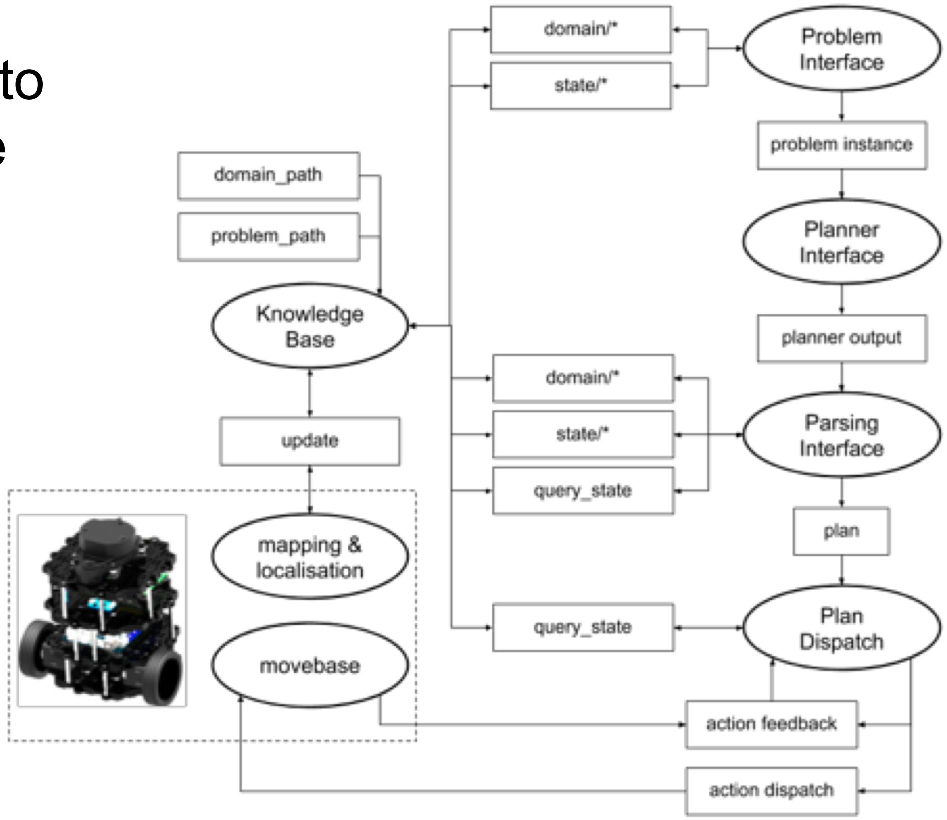
How to start with ROSPlan

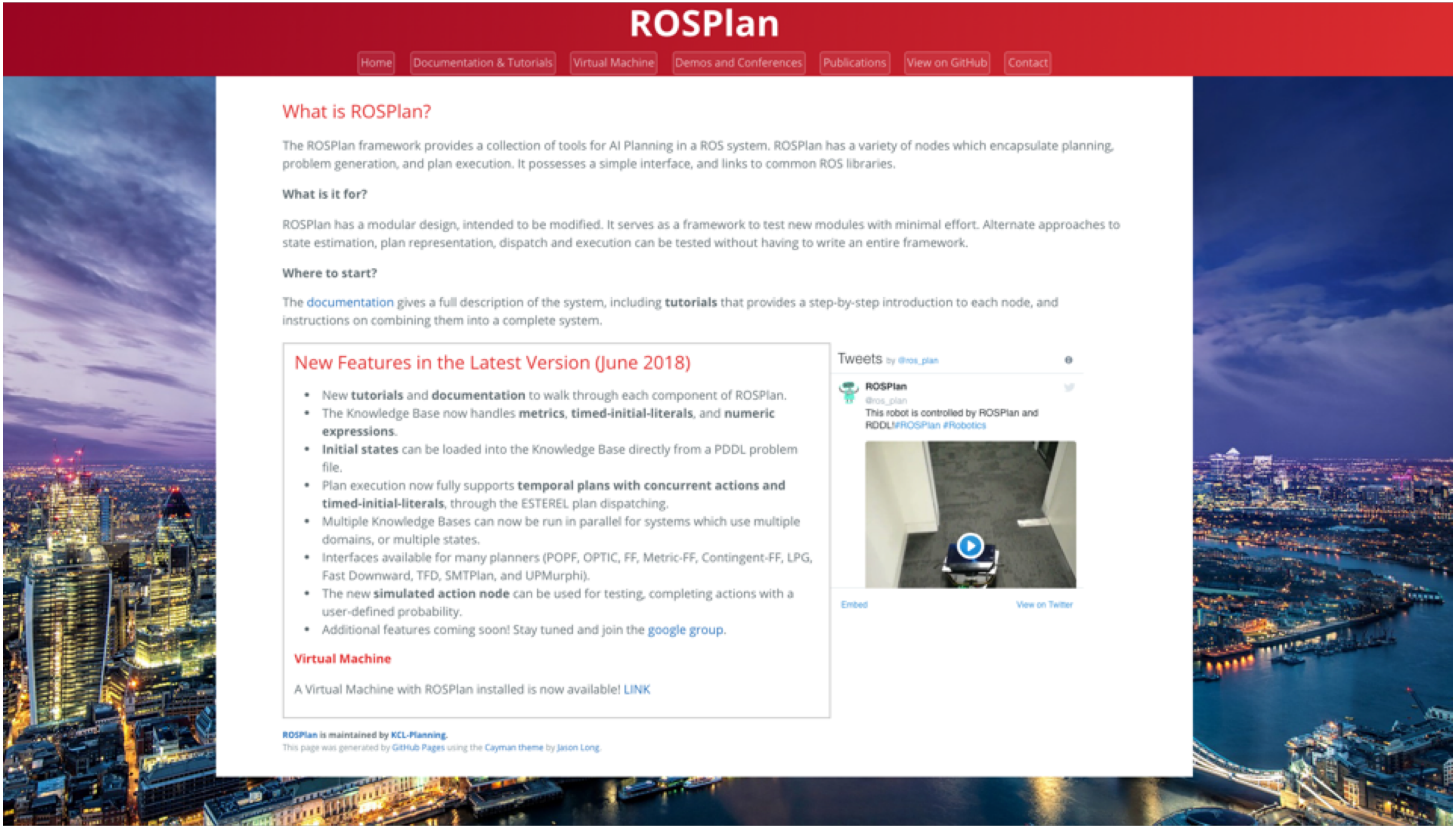
The default nodes can be combined to form a replanning system that is able to plan, execute plans, and replan when things go wrong.

Documentation and Tutorials:
kcl-planning.github.io/ROSPlan/

Open Source:
github.com/KCL-Planning/ROSPlan

Virtual Machine:
kcl-planning.github.io/ROSPlan/vm



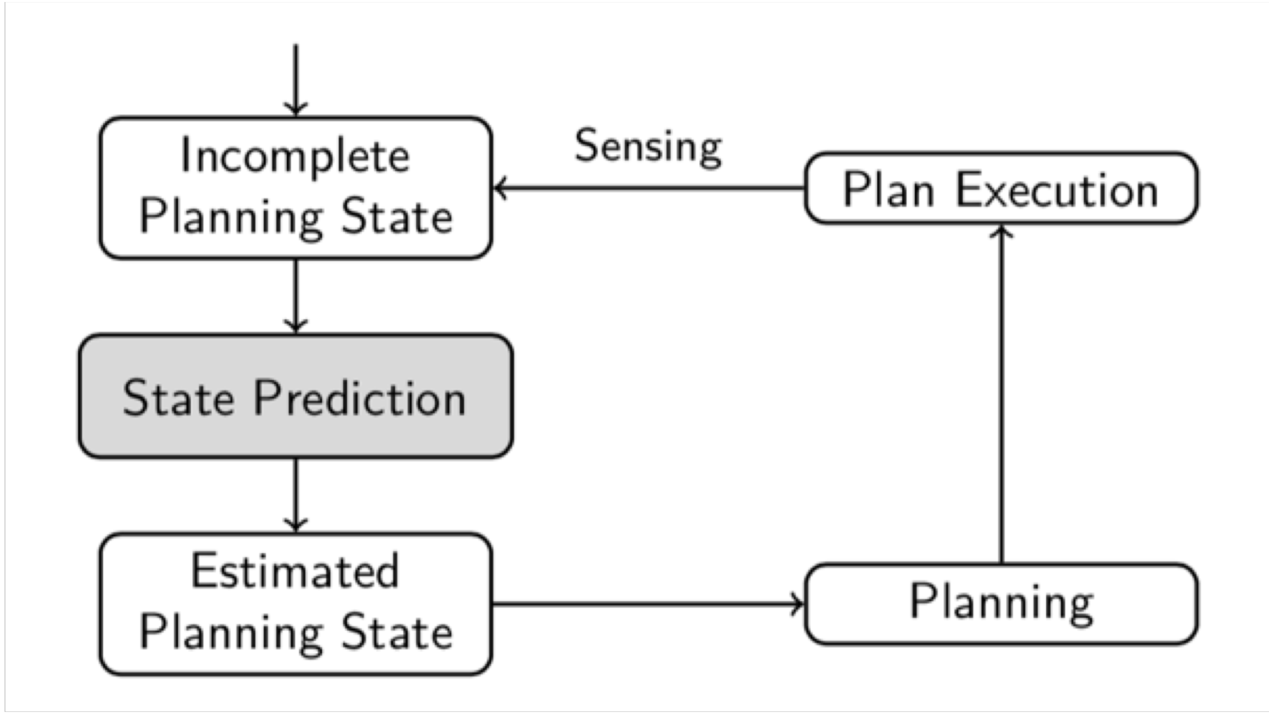


ROSPlan is open source: http://kcl-planning.github.io/ROSPlan/

AI Planning for Human-Robot Interaction

SQUIRREL
Clearing Clutter Bit by Bit

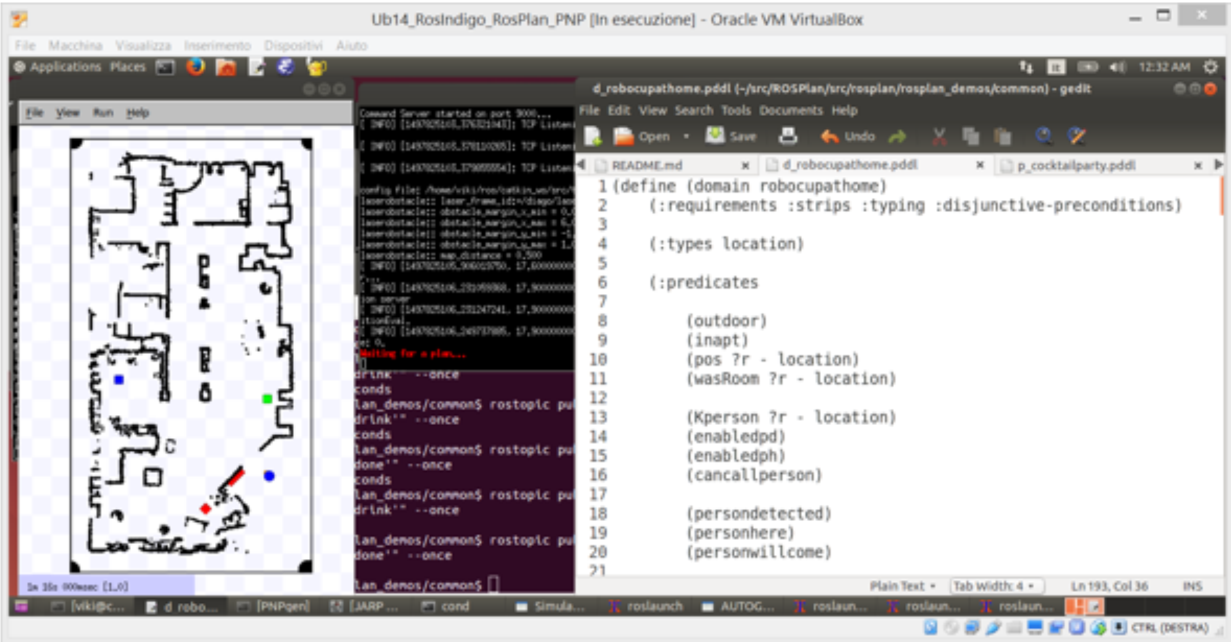
Decreasing State Uncertainty



Krivic, Cashmore, Magazzeni, Ridder, Szedmak, Piater. **Decreasing Uncertainty in Planning with State Prediction.** IJCAI 2017.

Planning for Human-Robot Interaction

When interacting with humans, plans can't be static
Conditional planning allows branches
Plans are dispatched as Petri-Nets and/or ESTEREL programs



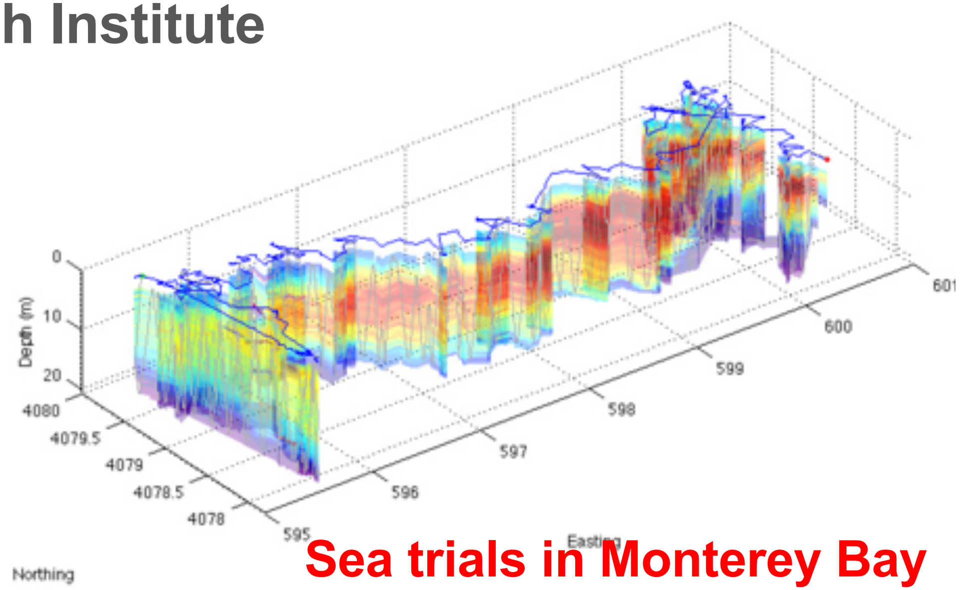
Sanelli, Cashmore, Magazzeni, Iocchi. **Short-Term Human Robot Interaction through Conditional Planning and Execution.**
ICAPS 2017.

AI Planning for Underwater Autonomy

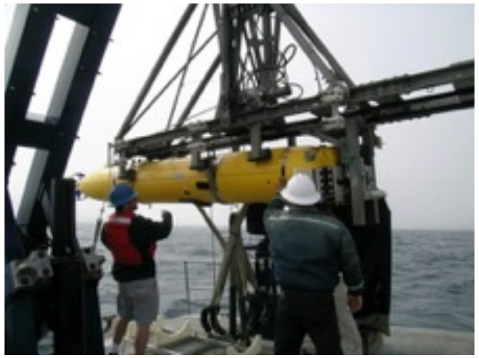
In collaboration with **Monterey Bay**

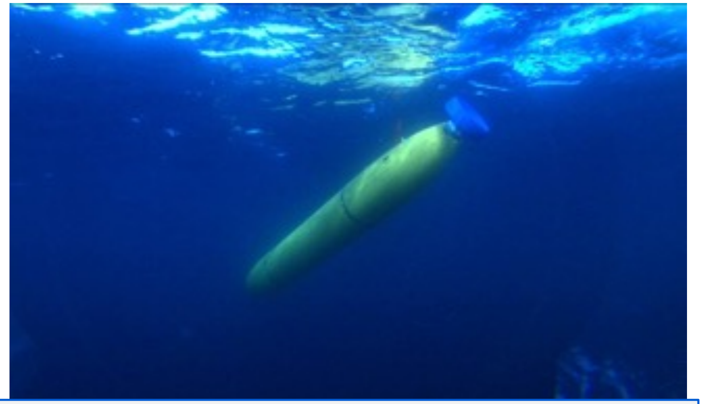
M B A R I **Aquarium Research Institute**

We used AI Planning for making AUVs autonomous in performing feature-tracking missions



Sea trials in Monterey Bay





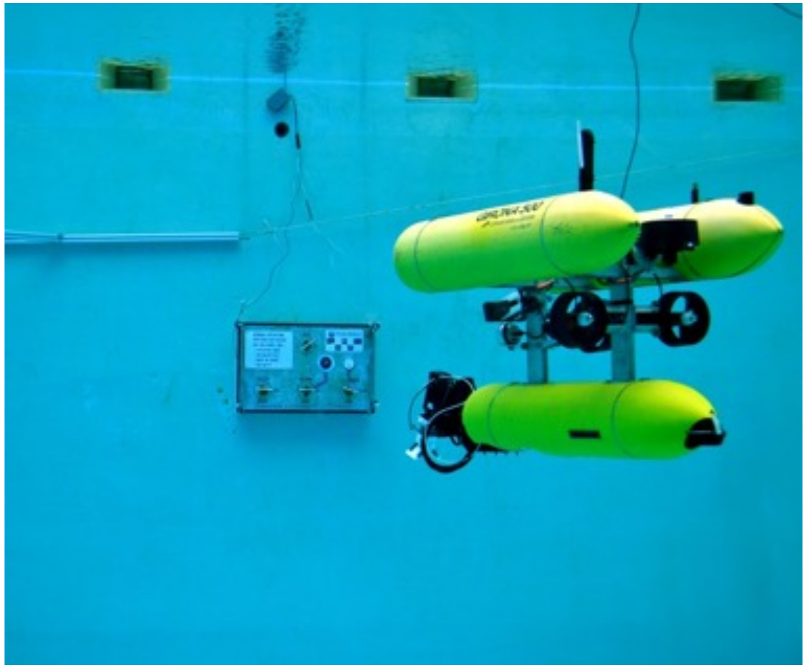
Magazzeni, Magazzeni, Py, Fox, Long, Rajan:. **DPolicy learning for autonomous feature tracking.** Autonomous Robots 37(1).

Autonomous Underwater Missions

Long-term maintenance and inspection of underwater oil installations
Persistent autonomy: planning, task learning, plan execution

Tasks:
- -inspect manifolds
- -clean manifolds
- -turn valves (time windows)
- -recharge AUV

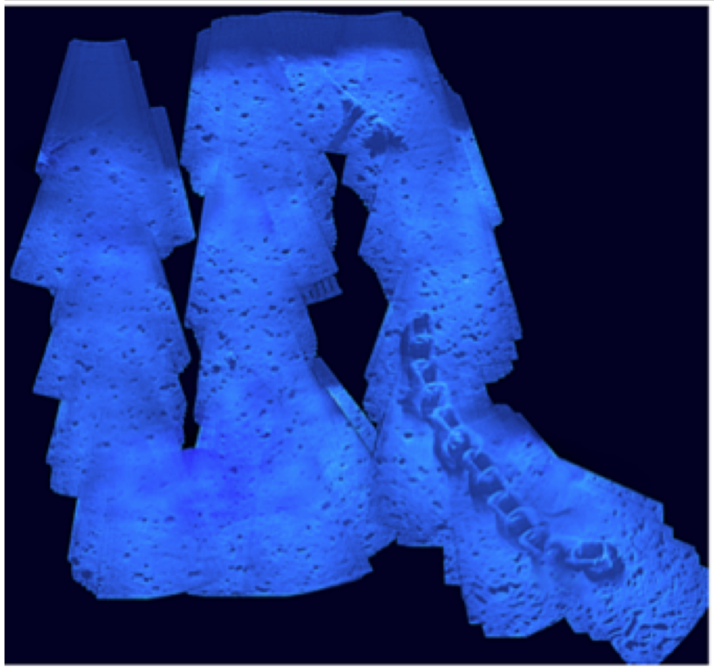
Pandora
Persistently Autonomous Robots

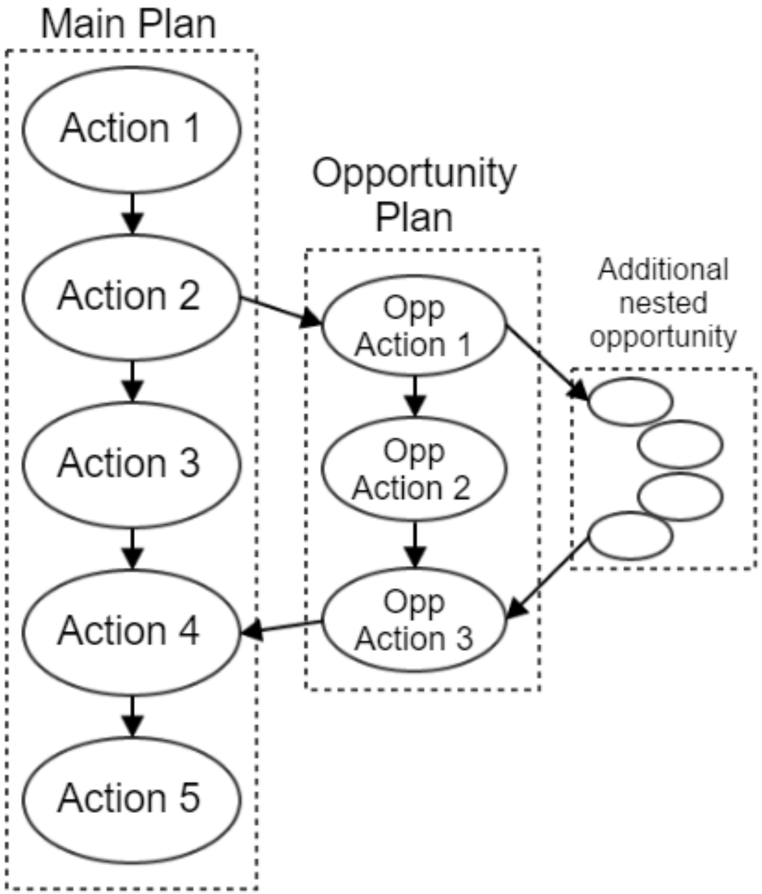


Girona 500 I-AUV
(ECA CSIP Manipulator)

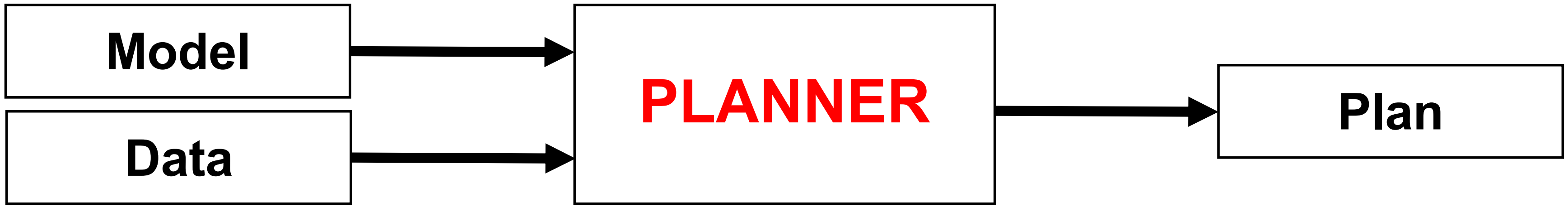
Opportunistic Planning

High-Impact-Low-Probability

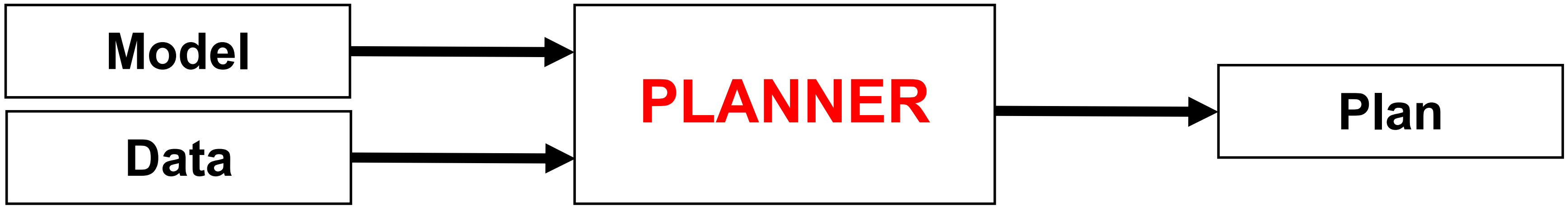




Cashmore, Fox, Long, Magazzeni, Ridder. **Opportunistic Planning in Autonomous Underwater Missions.**
IEEE Transactions on Automation Science and Engineering 15(2): 519-530 (2018)



ROSPlan



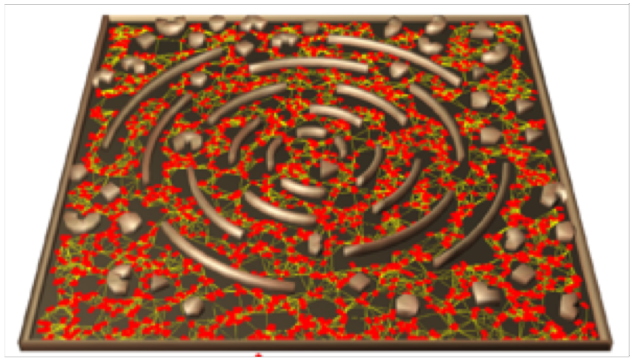
ROSPlan

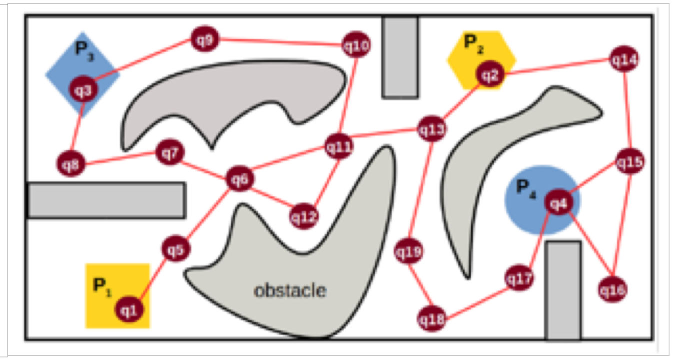
Task/Motion

Integrating Task/Motion Planning

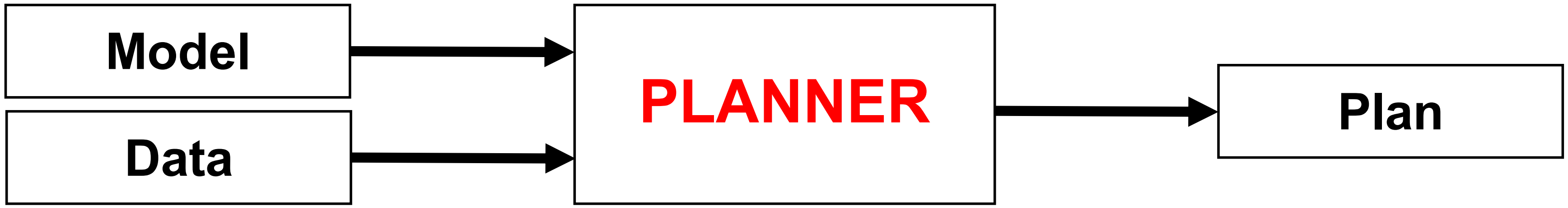
Decomposition into a discrete search and continuous motion plans.
Temporal planner considers waypoints for tasks in discrete space.
Sampling motion planner gives estimated duration for edges.
Temporal planner schedules motions and tasks to satisfy windows.
The planner reasons with tasks causality and preferences/priority.

Multi-Robots, Multi-Goals, Dynamics, Time Windows.



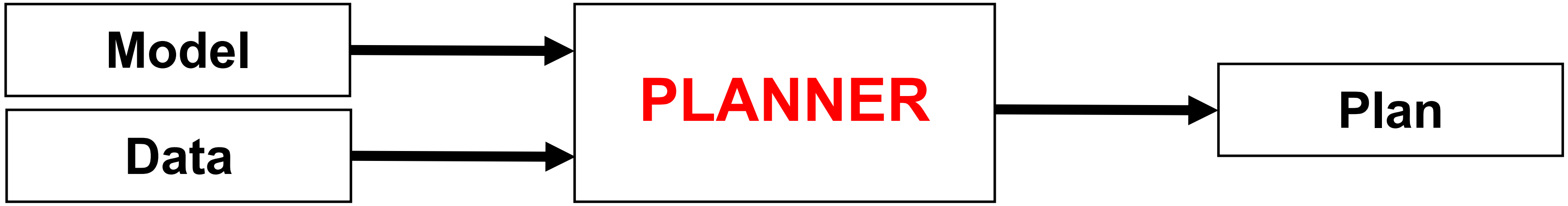


Edelkamp, Lahijanian, Magazzeni, Plaku. **Integrating Temporal Reasoning and Sampling-Based Motion Planning for Multi-Goal Problems with Dynamics and Time Windows.**
IROS 2018.



ROSPlan

Task/Motion



ROSPlan

Task/Motion

Strategic/ Tactical

Strategic/Tactical Planning

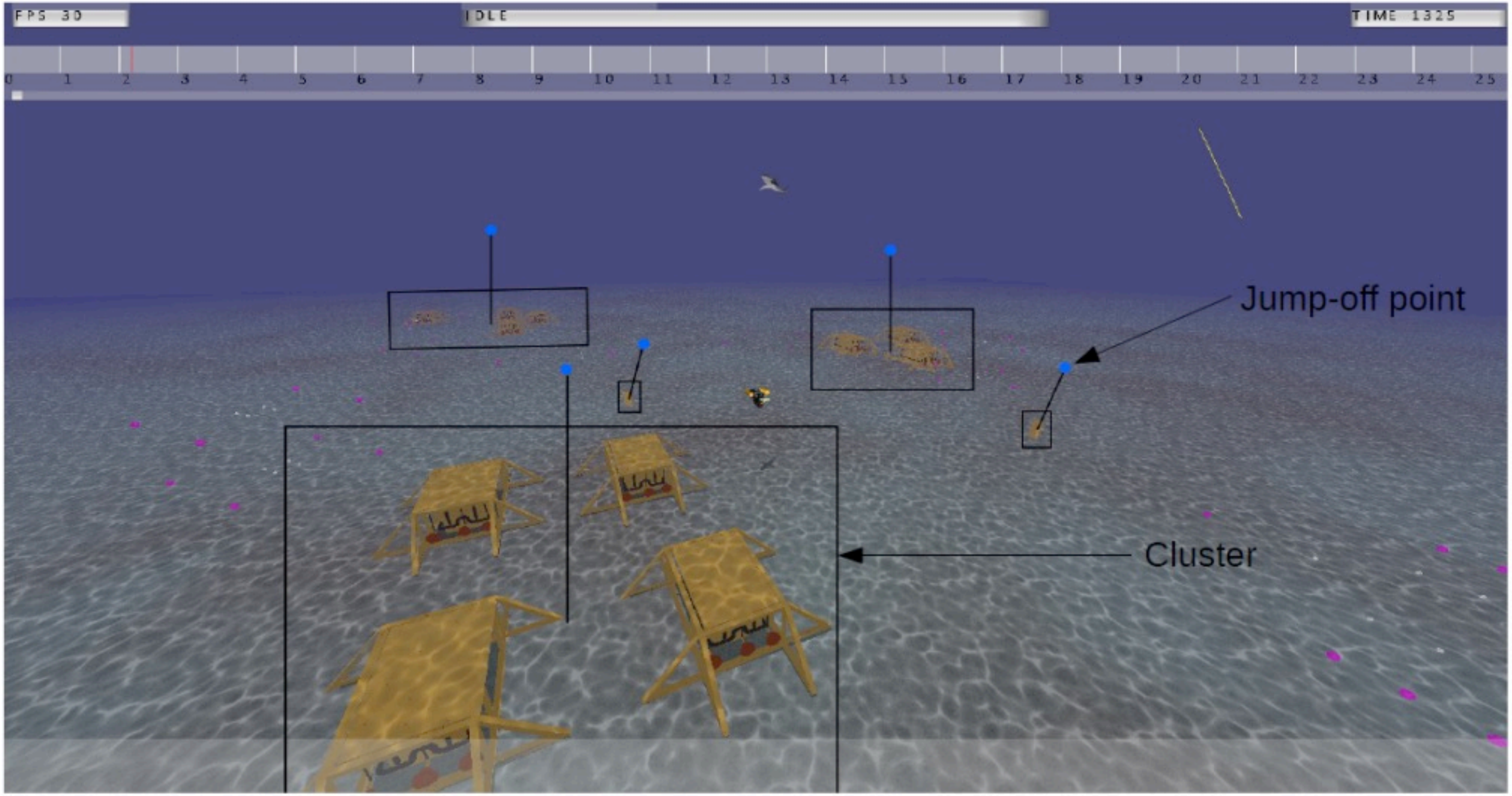
Cluster the *goals* into *tasks*

Strategic Layer: contains a high level plan that achieves all tasks and manages the resource and time constraints.

Tactical Layer: contains a plan that solves a single task.

Buksz, Cashmore, Krarup, Magazzeni. **Strategic-Tactical Planning for Autonomous Vehicles over Long Horizons.**
IROS 2018.

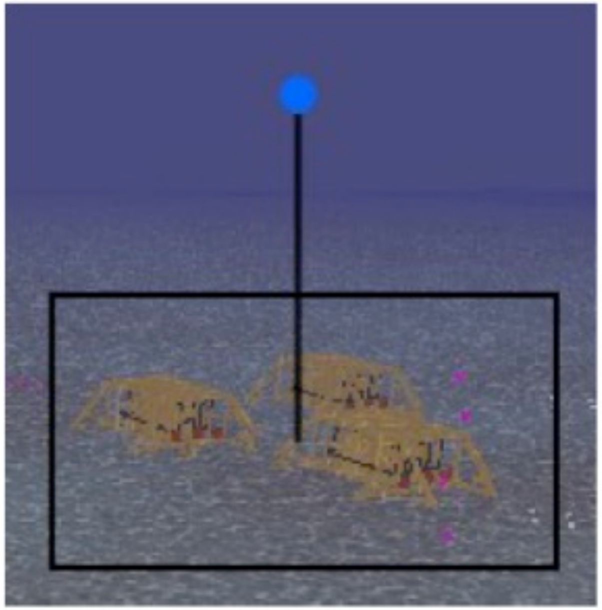
Strategic/Tactical Planning
Clustering



Strategic/Tactical Planning

Tactical Layer

For each Task the planner generates a plan and stores:
-duration
-resource constraints

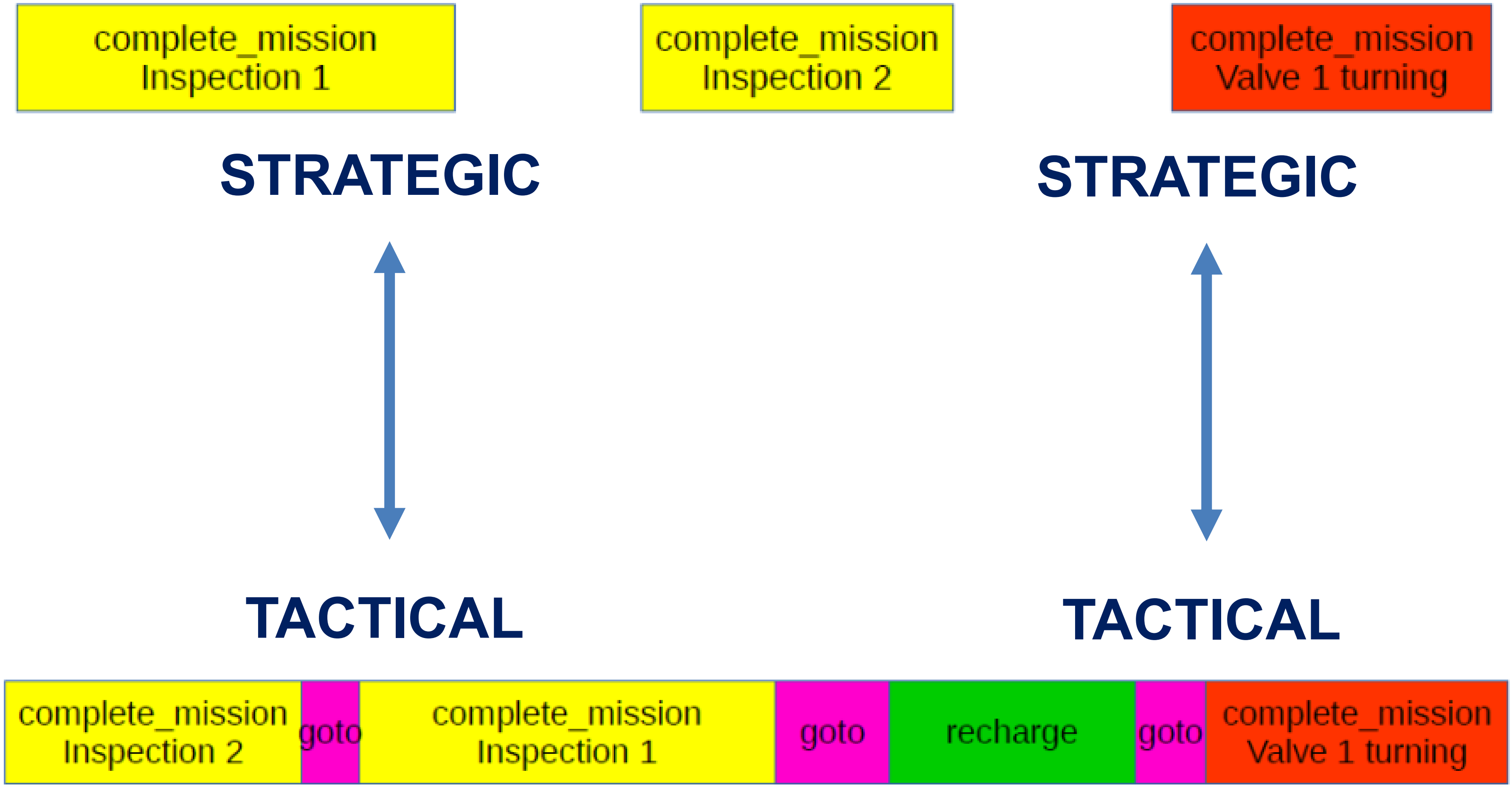


Energy consumption = 10W
Duration = 86.43s

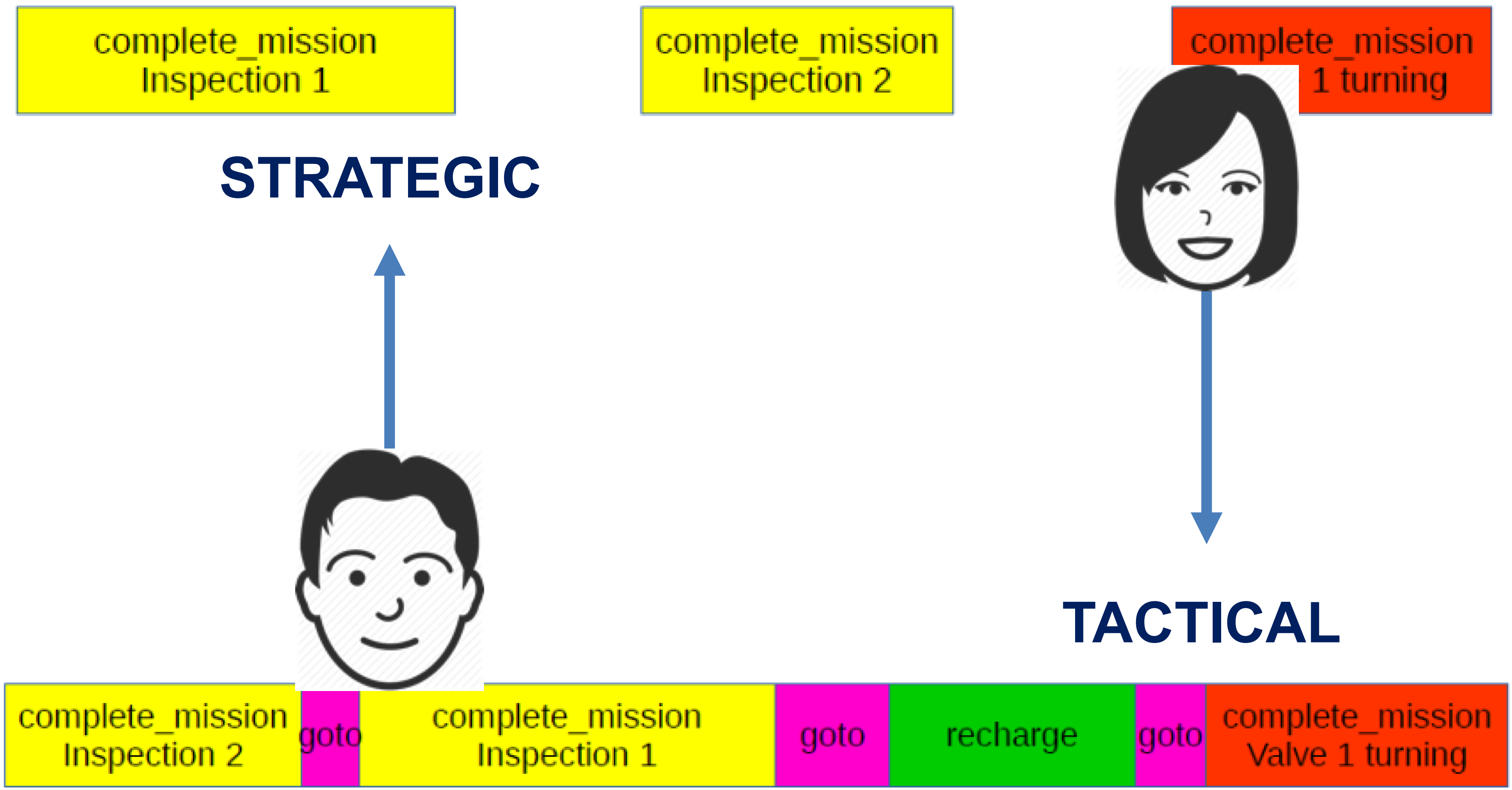
```
0.000: (correct_position auv0 wp_auv0) [3.000]
3.001: (do_hover_fast auv0 wp_auv0 strategic_location_7)
[11.403]
14.405: (correct_position auv0_strategic_location_78)
[3.000]
17.406: (observe_inspection_point auv0 strategic_location_7
inspection_point_2) [10.000]
27.407: (correct_position auv0 strategic_location_7)
[3.000]
45.083: (do_hover_controlled auv0 strategic_location_5
strategic_location_5) [4.000]
49.084: (observe_inspecetion_point auv0
strategic_location_5 inspection_point_4) [10.000]
...
```

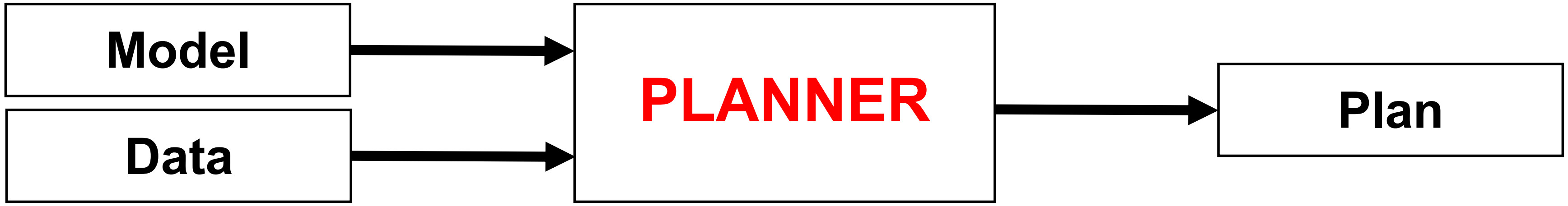
All the tactical plans are collected.

And the strategic plan is generated, not violating resource/time constraints



Now working on generalisation and human-AI teaming





ROSPlan

Task/Motion

Strategic/Tactical

Trust in Autonomous Systems

Main obstruction to deployment of Autonomous Systems:

lack of **trust**

For the humans there is insufficient understanding in the underlying AI processes that govern the autonomous systems, which become **black boxes** to the user.

In order to engender trust, humans must understand what the AI system is trying to achieve, and why.

Explainable AI

General Data Protection Regulation

Article 12: Transparent information, communication and modalities for the exercise of the rights of the data subject

The controller shall take appropriate measures to provide any information referred to in Articles 13 and 14 and any communication under Articles 15 to 22 and 34 relating to processing to the data subject in a **concise**, **transparent**, **intelligible** and **easily accessible form**, using clear and plain language.

Article 13: Information to be provided where personal data are collected from the data subject

The controller shall provide […] the existence of automated decision-making, including **meaningful information about the logic involved**, as well as the significance and the envisaged consequences of such processing for the data subject.

IJCAI 2019 WORKSHOP ON EXPLAINABLE ARTIFICIAL INTELLIGENCE (XAI)

10, 11, or 12 August, 2019. Macau, China

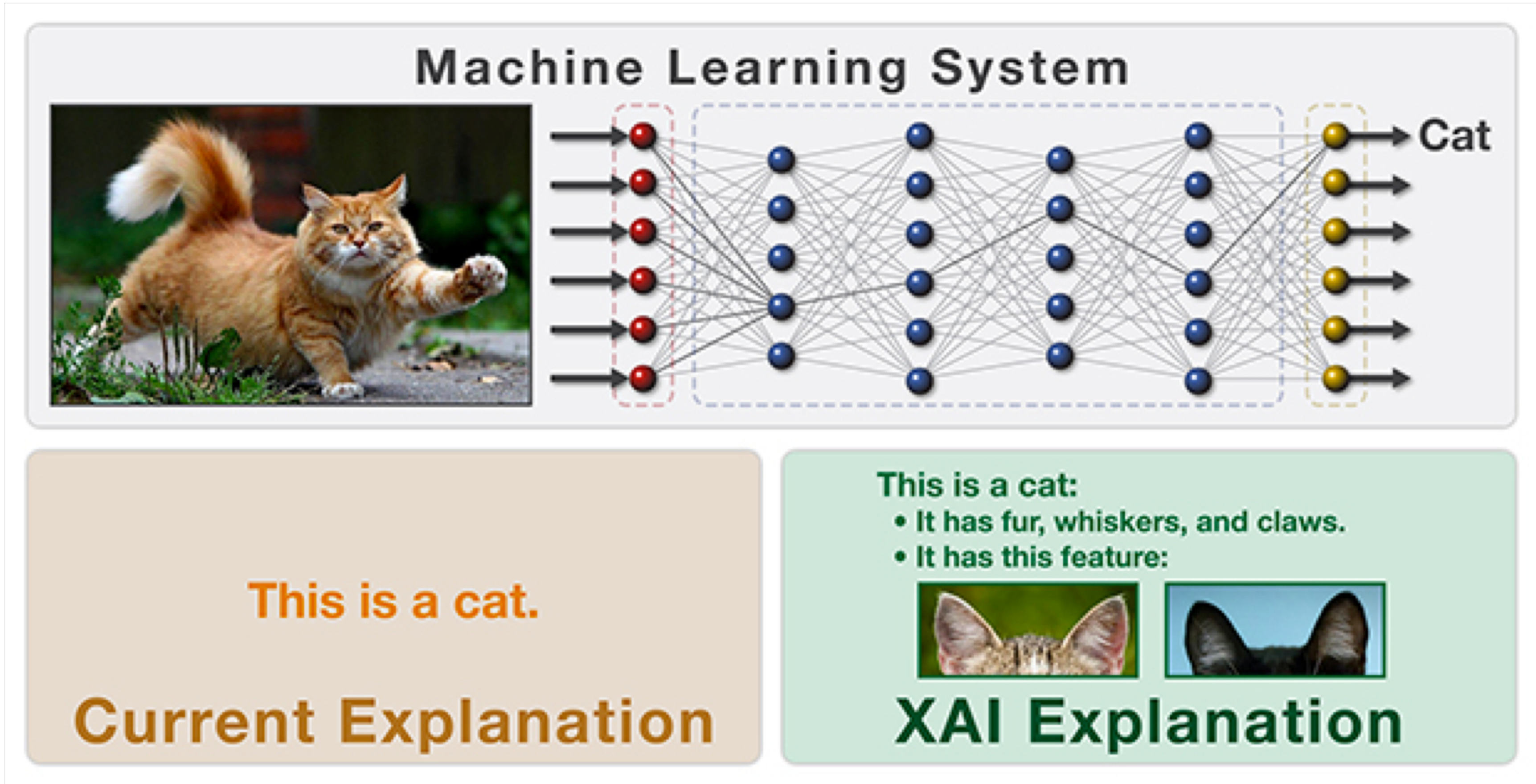
https://www.ijcai19.org/

Explainable AI

Co-Chairs:
David Aha (NRL, USA)
Daniele Magazzeni (King’s College London)
Tim Miller (University of Melbourne, Australia)
Rosina Weber (Drexel University)

Submission deadline: 19 May 2019

Explainable AI



Data-Driven AI

Attentive Explanations: Justifying Decisions and Pointing to the Evidence

Dong Huk Park[1] Lisa Anne Hendricks[1] Zeynep Akata[1,2]
Bernt Schiele[2] Trevor Darrell[1] Marcus Rohrbach[1]

[1]UC Berkeley EECS, CA, United States
[2]Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

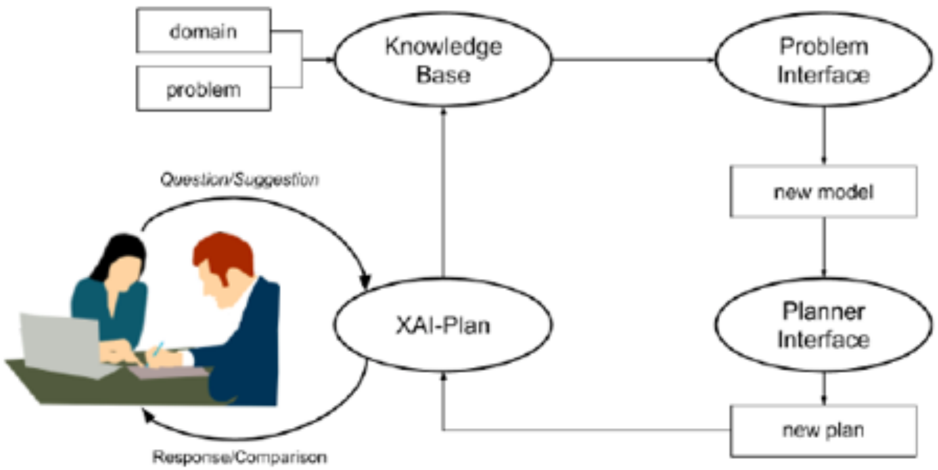
On the Use of Opinionated Explanations to Rank and Justify Recommendations

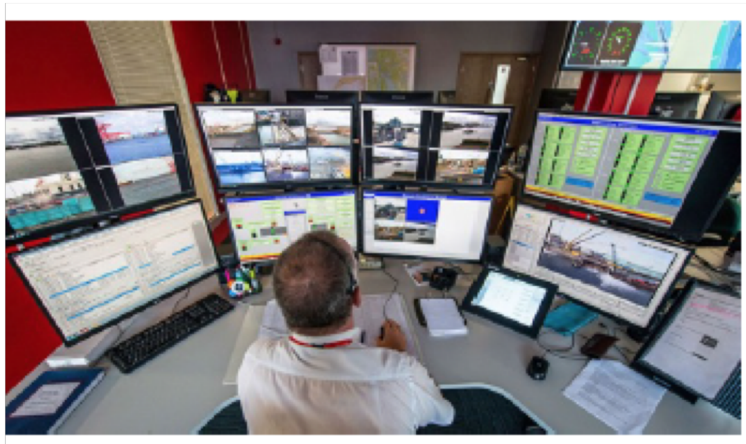
Khalil Muhammad, Aonghus Lawlor, Barry Smyth

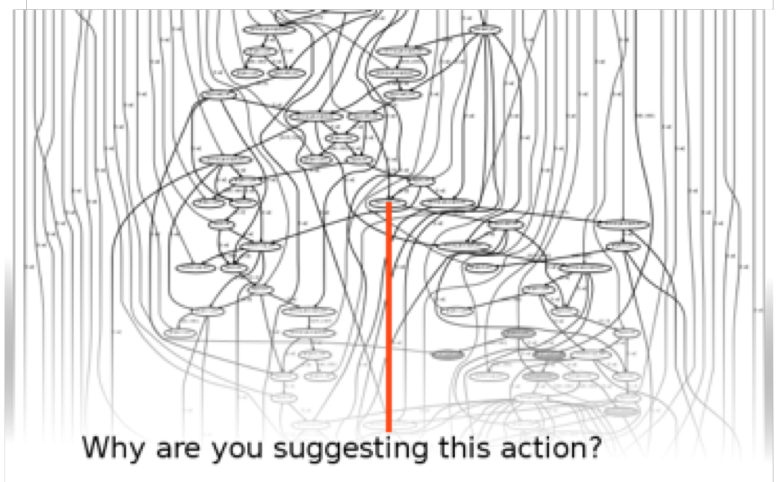
Insight Centre for Data Analytics
University College Dublin
Belfield, Dublin 4, Ireland
{khalil.muhammad, aonghus.lawlor, barry.smyth}@insight-centre.org

Explainable AI Planning (XAIP)

- **Need for Trust, Interaction, and Transparency**
- **Human operators (especially those in charge of /responsible for critical decisions) want to understand why the AI suggests something that they would not do.**
- **Intelligent Situational Awareness.**







(*some*) Things to Be Explained

- Q1: Why did you do that?
- Q2: Why didn't you do *something else*? (that I would have done)
- Q3: Why is what you propose to do more efficient/safe/cheap than something else? (that I would have done)
- Q4: Why can't you do that ?
- Q5: Why do I need to replan at this point?
- Q6: Why do I not need to replan at this point?

Fox, Long, Magazzeni. **Explainable Planning.**
IJCAI 2017 Workshop on Explainable AI.

Providing Explanations

- Q2: Why didn't you do *something else*? (that I would have done)

Quick (*and useless*) answer: because the heuristic evaluation was better for the decision the planner made.

We should demonstrate that the alternative action would prevent from finding a valid plan or would lead to a plan that is no better than the one found by the planner.

Contrastive Explanations

Providing Explanations

- Q2: Why didn't you do *something else*? (that I would have done)

Algorithm:

-re-run the planner up to the decision point questioned by the human

-inject the human choice

-plan from the state obtained after applying the action chosen by the human

Providing Explanations

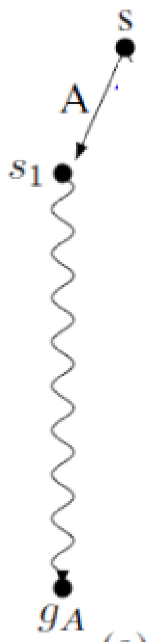
- Q2: Why didn't you do *something else*? (that I would have done)

Algorithm:

-re-run the planner up to the decision point questioned by the human

-inject the human choice

-plan from the state obtained after applying the action chosen by the human



(a)

Providing Explanations

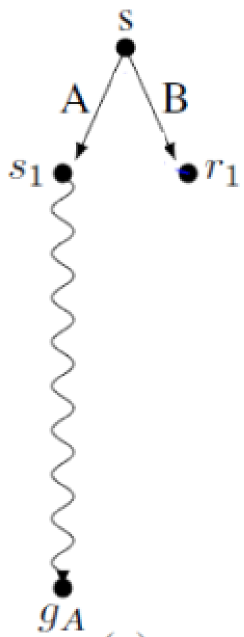
- Q2: Why didn't you do *something else*? (that I would have done)

Algorithm:

-re-run the planner up to the decision point questioned by the human

-inject the human choice

-plan from the state obtained after applying the action chosen by the human



(a)

Providing Explanations

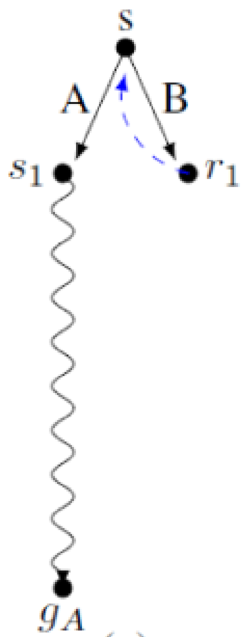
- Q2: Why didn't you do *something else*? (that I would have done)

Algorithm:

-re-run the planner up to the decision point questioned by the human

-inject the human choice

-plan from the state obtained after applying the action chosen by the human



(a)

Providing Explanations

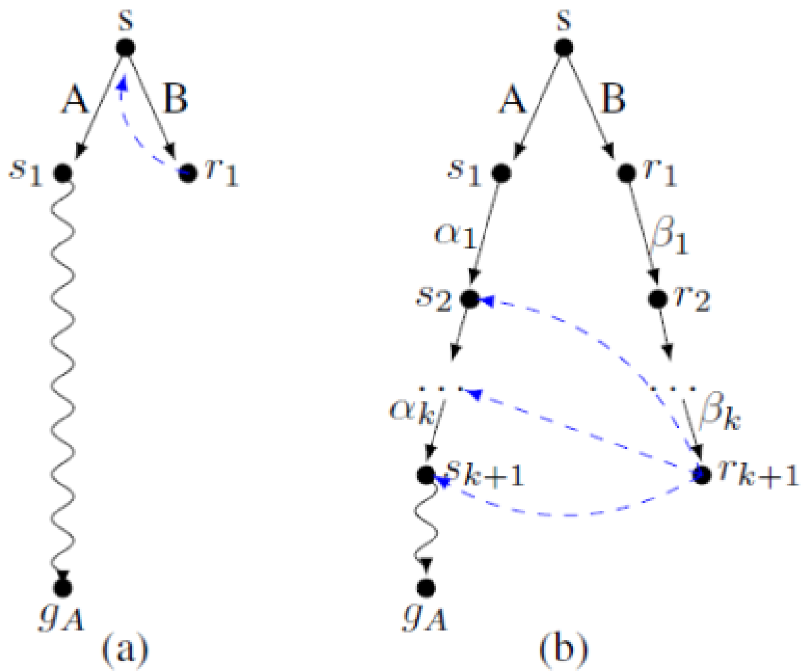
- Q2: Why didn't you do *something else*? (that I would have done)

Algorithm:

-re-run the planner up to the decision point questioned by the human

-inject the human choice

-plan from the state obtained after applying the action chosen by the human



Providing Explanations

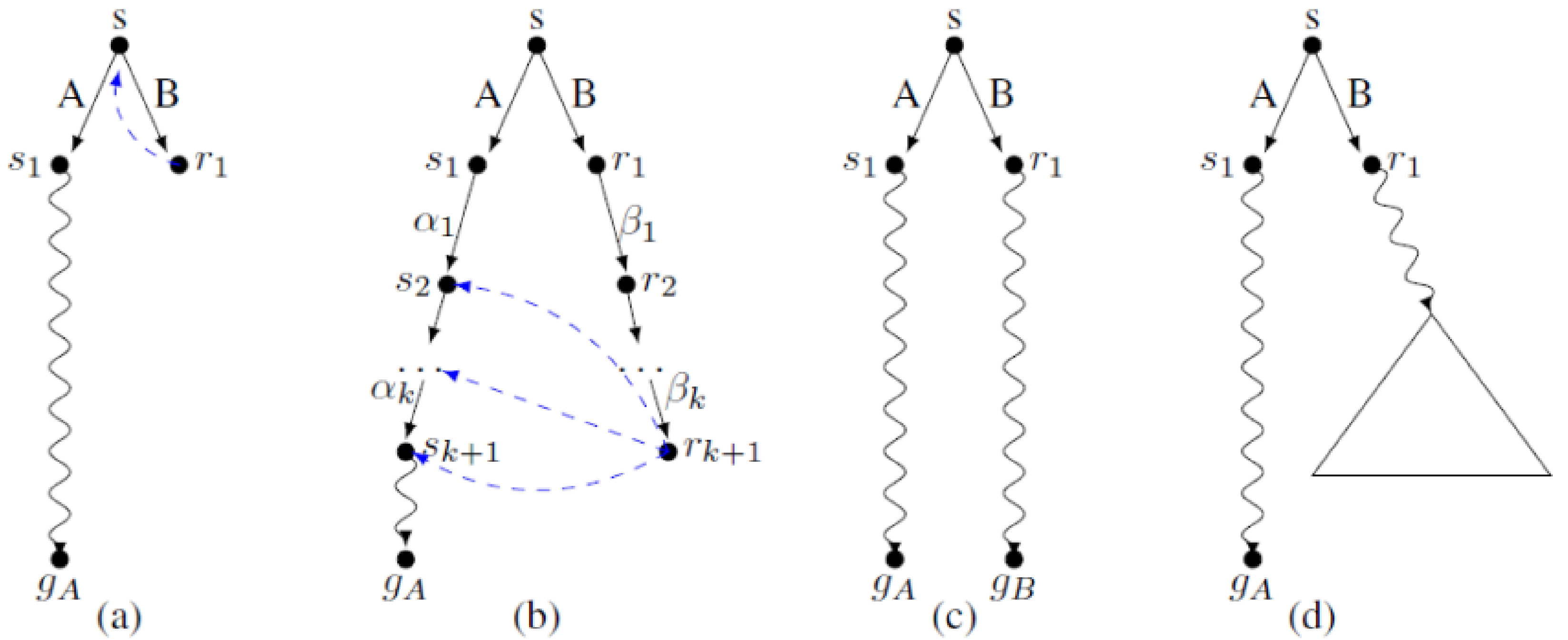
- Q2: Why didn't you do *something else*? (that I would have done)

Algorithm:

-re-run the planner up to the decision point questioned by the human

-inject the human choice

-plan from the state obtained after applying the action chosen by the human



Illustrative Example

`Rover Time` domain from IPC-4 (problem 3)

```
0.000: (navigate r1 wp3 wp0)   [5.0]
0.000: (navigate r0 wp1 wp0)   [5.0]
5.001: (calibrate r1 camera1 obj0 wp0)   [5.0]
5.001: (sample_rock r0 r0store wp0)   [8.0]
10.002: (take_image r1 wp0 obj0 camera1 col)   [7.0]
13.001: (navigate r0 wp0 wp1)   [5.0]
17.002: (navigate r1 wp0 wp3)   [5.0]
18.001: (comm_rock_data r0 general wp0 wp1 wp0)   [10.0]
22.003: (navigate r1 wp3 wp2)   [5.0]
27.003: (sample_soil r1 r1store wp2)   [10.0]
28.002: (comm_image_data r1 general obj0 col wp2 wp0) [15.0]
43.003: (comm_soil_data r1 general wp2 wp2 wp0)   [10.0]

[Duration = 53.003]
```

Q1: *why did you use Rover0 to take the rock sample at waypoint0* ?

NA: *so that I can communicate_data from Rover0 later (at 18.001)*

Illustrative Example

`Rover Time` domain from IPC-4 (problem 3)

```
0.000: (navigate r1 wp3 wp0)   [5.0]
0.000: (navigate r0 wp1 wp0)   [5.0]
5.001: (calibrate r1 camera1 obj0 wp0)   [5.0]
5.001: (sample_rock r0 r0store wp0)   [8.0]
10.002: (take_image r1 wp0 obj0 camera1 col)   [7.0]
13.001: (navigate r0 wp0 wp1)   [5.0]
17.002: (navigate r1 wp0 wp3)   [5.0]
18.001: (comm_rock_data r0 general wp0 wp1 wp0)   [10.0]
22.003: (navigate r1 wp3 wp2)   [5.0]
27.003: (sample_soil r1 r1store wp2)   [10.0]
28.002: (comm_image_data r1 general obj0 col wp2 wp0) [15.0]
43.003: (comm_soil_data r1 general wp2 wp2 wp0)   [10.0]

[Duration = 53.003]
```

Q1: *why did you use Rover0 to take the rock sample at waypoint0* ?

why didn't Rover1 take the rock sample at waypoint0 ?

Illustrative Example

Q1: *why did you use Rover0 to take the rock sample at waypoint0 ?*

why didn't Rover1 take the rock sample at waypoint0 ?

We remove the ground action instance for Rover0 and re-plan

A: ***Because not using Rover0 for this action leads to a longer plan***

```
0.000: (navigate r1 wp3 wp0)   [5.0]
5.001: (calibrate r1 camera1 obj0 wp0)   [5.0]
10.002: (take_image r1 wp0 obj0 camera1 col)   [7.0]
10.003: (sample_rock r1 r1store wp0)   [8.0]
18.003: (navigate r1 wp0 wp3)   [5.0]
18.004: (drop r1 r1store)   [1.0]
23.004: (navigate r1 wp3 wp2)   [5.0]
28.004: (comm_image_data r1 general obj0 col wp2 wp0) [15.0]
28.005: (sample_soil r1 r1store wp2)   [10.0]
43.005: (comm_soil_data r1 general wp2 wp2 wp0)   [10.0]
53.006: (comm_rock_data r1 general wp0 wp2 wp0)   [10.0]

[Duration = 63.006]
```

```
0.000: (naviga
0.000: (naviga
5.001: (calibr
5.001: (sample
10.002: (take_
13.001: (navig
17.002: (navig
18.001: (comm_
22.003: (navig
27.003: (sampl
28.002: (comm_image_data r1 general obj0 col wp2 wp0) [15.0]
43.003: (comm_soil_data r1 general wp2 wp2 wp0)   [10.0]

[Duration = 53.003]
```

Illustrative Example

Q1: *why did you use Rover0 to take the rock sample at waypoint0* ?

***why didn't Rover1 take the rock sample at waypoint0* ?**

We remove the ground action instance for Rover0 and re-plan

A: *Because not using Rover0 for this action leads to a longer plan*

Q2: But why does Rover1 do everything in this plan?

```
0.000: (navigate r1 wp3 wp0)  [5.0]
5.001: (calibrate r1 camera1 obj0 wp0)  [5.0]
10.002: (take_image r1 wp0 obj0 camera1 col)  [7.0]
10.003: (sample_rock r1 r1store wp0)  [8.0]
18.003: (navigate r1 wp0 wp3)  [5.0]
18.004: (drop r1 r1store)  [1.0]
23.004: (navigate r1 wp3 wp2)  [5.0]
28.004: (comm_image_data r1 general obj0 col wp2 wp0) [15.0]
28.005: (sample_soil r1 r1store wp2)  [10.0]
43.005: (comm_soil_data r1 general wp2 wp2 wp0)  [10.0]
53.006: (comm_rock_data r1 general wp0 wp2 wp0)  [10.0]

[Duration = 63.006]
```

Illustrative Example

Q1: *why did you use Rover0 to take the rock sample at waypoint0* ?

***why didn't Rover1 take the rock sample at waypoint0* ?**

We remove the ground action instance for Rover0 and re-plan

A: *Because not using Rover0 for this action leads to a longer plan*

***Q2: But why does Rover1 do everything in this plan*?**

We require the plan to contain at least one action that has Rover0 as argument (add dummy effect to all actions using Rover0 and put into the goal)

```
0.000:  (na
5.001:  (ca
10.002:  (t
10.003:  (s
18.003:  (r
18.004:  (c
23.004:  (r
28.004:  (c
28.005:  (s
43.005:  (c
53.006:  (c

[Duration
```

```
0.000:  (navigate r0 wp1 wp0)   [5.0]
0.000:  (navigate r1 wp3 wp0)   [5.0]
5.001:  (calibrate r1 camera1 obj0 wp0)   [5.0]
10.002:  (take_image r1 wp0 obj0 camera1 col)   [7.0]
10.003:  (sample_rock r1 r1store wp0)   [8.0]
18.003:  (navigate r1 wp0 wp3)   [5.0]
18.004:  (drop r1 r1store)   [1.0]
23.004:  (navigate r1 wp3 wp2)   [5.0]
28.004:  (comm_image_data r1 general obj0 col wp2 wp0)   [15.0]
28.005:  (sample_soil r1 r1store wp2)   [10.0]
43.005:  (comm_soil_data r1 general wp2 wp2 wp0)   [10.0]
53.006:  (comm_rock_data r1 general wp0 wp2 wp0)   [10.0]
```

Illustrative Example

Q1: *why did you use Rover0 to take the rock sample at waypoint0* ?

***why didn't Rover1 take the rock sample at waypoint0* ?**

We remove the ground action instance for Rover0 and re-plan

A: *Because not using Rover0 for this action leads to a longer plan*

***Q2: But why does Rover1 do everything in this plan*?**

We require the plan to contain at least one action that has Rover0 as argument (add dummy effect to all actions using Rover0 and put into the goal)

A: *There is no useful way to use Rover0 for improving this plan*

```
0.000: (navigate r0 wp1 wp0)   [5.0]
0.000: (navigate r1 wp3 wp0)   [5.0]
5.001: (calibrate r1 camera1 obj0 wp0)   [5.0]
10.002: (take_image r1 wp0 obj0 camera1 col)   [7.0]
10.003: (sample_rock r1 r1store wp0)   [8.0]
18.003: (navigate r1 wp0 wp3)   [5.0]
18.004: (drop r1 r1store)   [1.0]
23.004: (navigate r1 wp3 wp2)   [5.0]
28.004: (comm_image_data r1 general obj0 col wp2 wp0)   [15.0]
28.005: (sample_soil r1 r1store wp2)   [10.0]
43.005: (comm_soil_data r1 general wp2 wp2 wp0)   [10.0]
53.006: (comm_rock_data r1 general wp0 wp2 wp0)   [10.0]
```

Illustrative Example

Q1: *why did you use Rover0 to take the rock sample at waypoint0* ?

***why didn't Rover1 take the rock sample at waypoint0* ?**

We remove the ground action instance for Rover0 and re-plan

A: *Because not using Rover0 for this action leads to a longer plan*

Q2: But why does Rover1 do everything in this plan?

We require the plan to contain at least one action that has Rover0 as argument (add dummy effect to all actions using Rover0 and put into the goal)

A: *There is no useful way to use Rover0 for improving this plan*

Q3: Can't you use both Rover0 and Rover1 to achieve the goal?

We restrict the actions that achieve the dummy condition to the set of actions that achieve the actual goals

Illustrative Example

Q1: *why did you use Rover0 to take the rock sample at waypoint0* ?

***why didn't Rover1 take the rock sample at waypoint0* ?**

We remove the ground action instance for Rover0 and re-plan

A: *Because not using Rover0 for this action leads to a longer plan*

Q2: But why does Rover1 do everything in this plan?

We require the plan to contain at least one action that has Rover0 as argument (add dummy effect to all actions using Rover0 and put into the goal)

A: *There is no useful way to use Rover0 for improving this plan*

Q3: Can't you use both Rover0 and Rover1 to achieve the goal?

We restrict the actions that achieve the dummy condition to the set of actions that achieve the actual goals

`Plan not found !`

Providing Explanations

- Q3: Why what you want to do is more efficient/safe/cheap than something else? (that I would do)

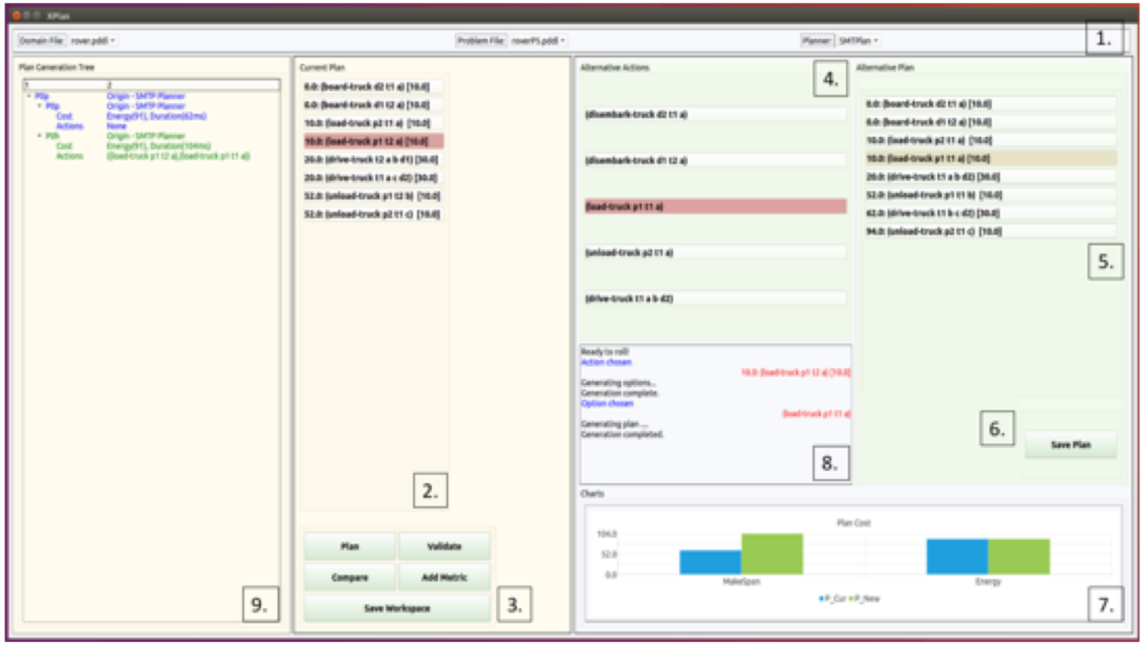
Different metrics can be used to evaluate the plan.

For complex domains, most planners can only optimise makespan, but not other metrics.

Combine planners with the plan validator

VAL allows the evaluation of plans using different metrics

Explainable AI Planning (XAIP)



XAI-Plan is a general framework that can be *customised* to specific users/scenarios (taking into account their modus operandi, languages, preferences, situational awareness factors, etc).

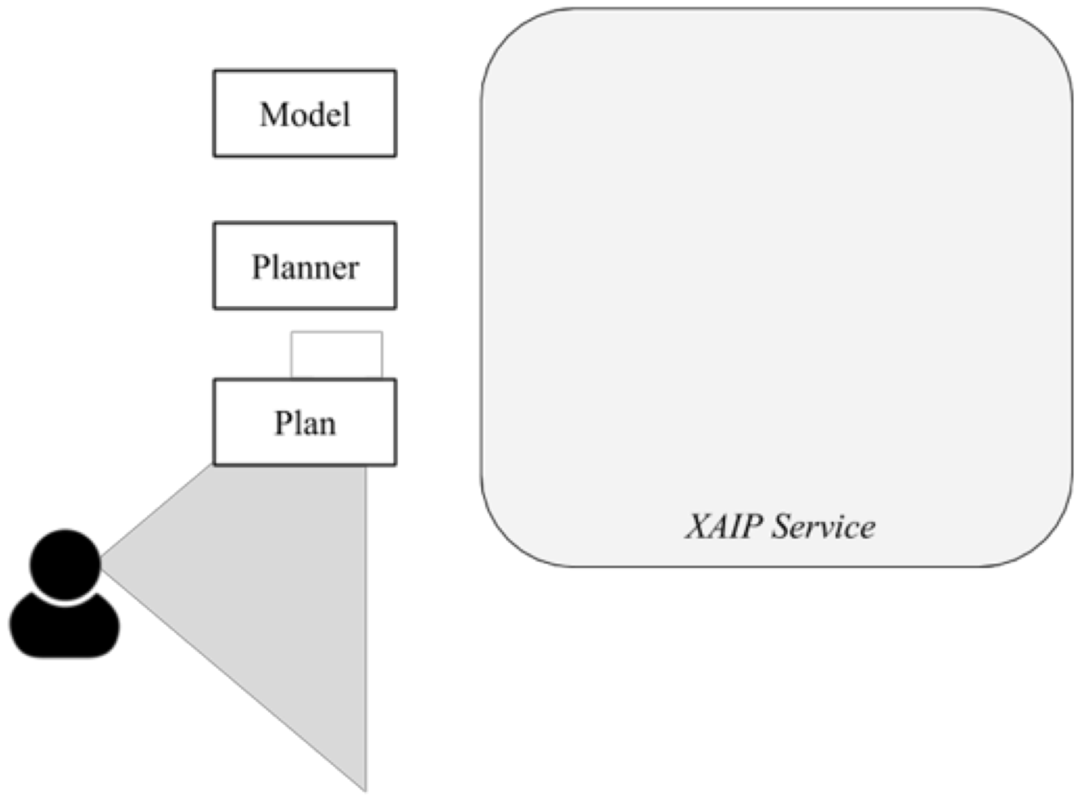
Borgo, Cashmore, Magazzeni. **Towards Providing Explanations for Planner Decisions.**
IJCAI 2018.

Explainable Planning as a Service

1. **The supervisor will not accept an explanation generated by a planner different from the one that they use and whose performance they trust.**

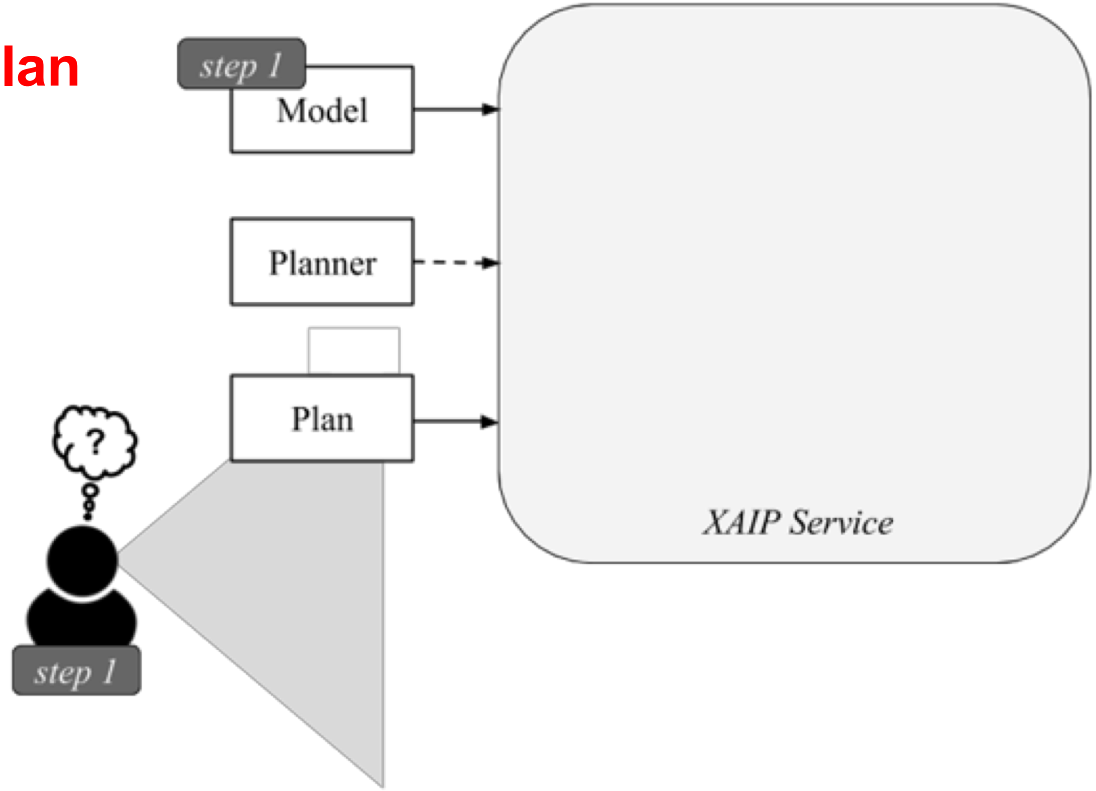
2. **The supervisor will not accept an explanation generated using a model that differs from the one that has been developed by the company's engineers, verified, and is trusted by the supervisor.**

Explainable Planning as a Service



Explainable Planning as a Service

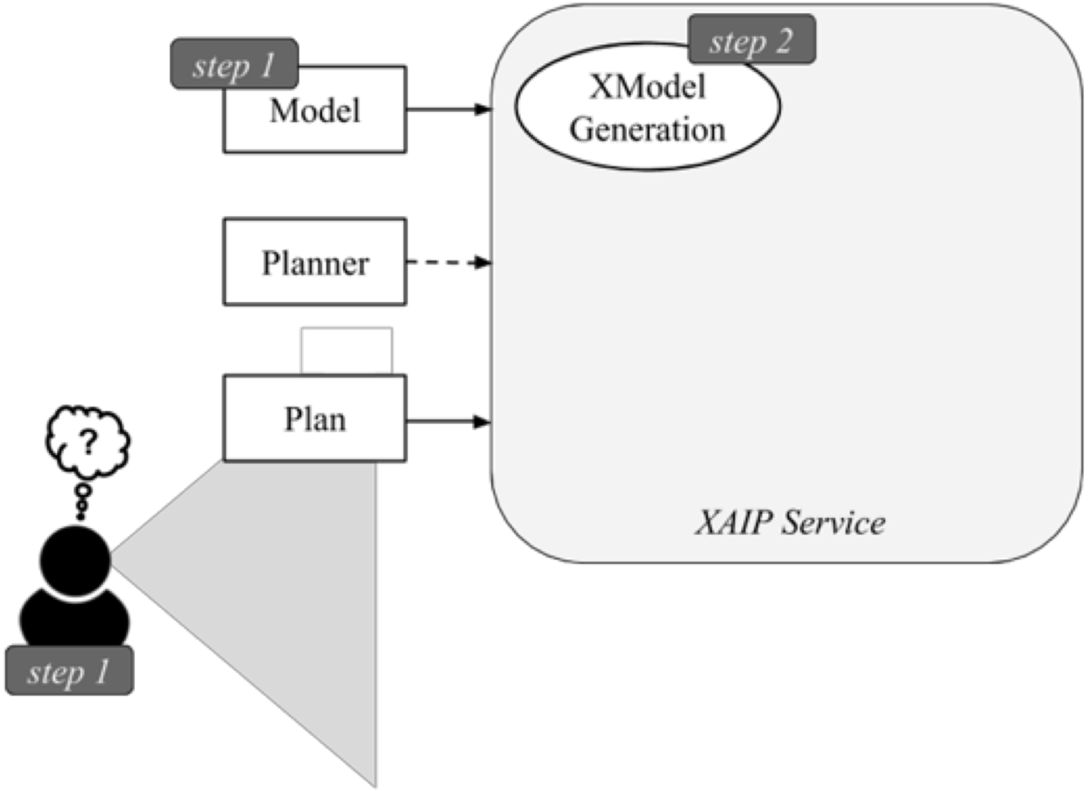
Step 1: Questioning the plan



The XAIP Service takes as input:
the model, the plan, and the question from the user

Explainable Planning as a Service

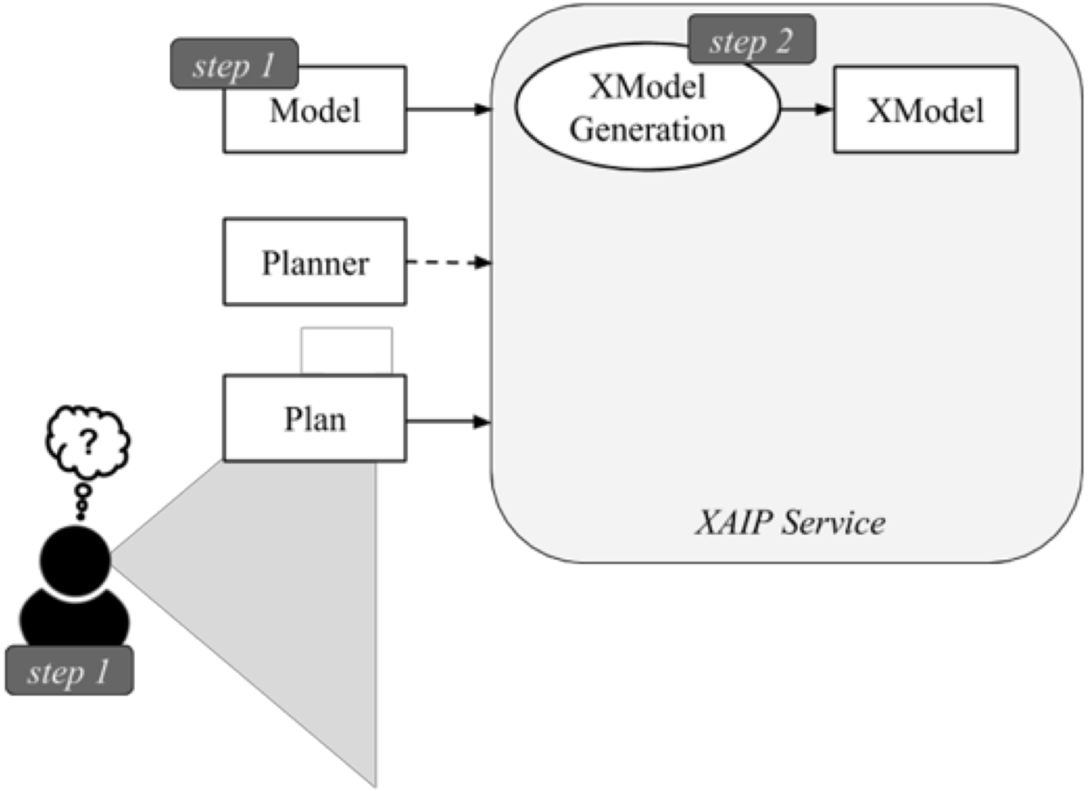
Step 2: Deriving the Explanation Model



The query is translated into constraints

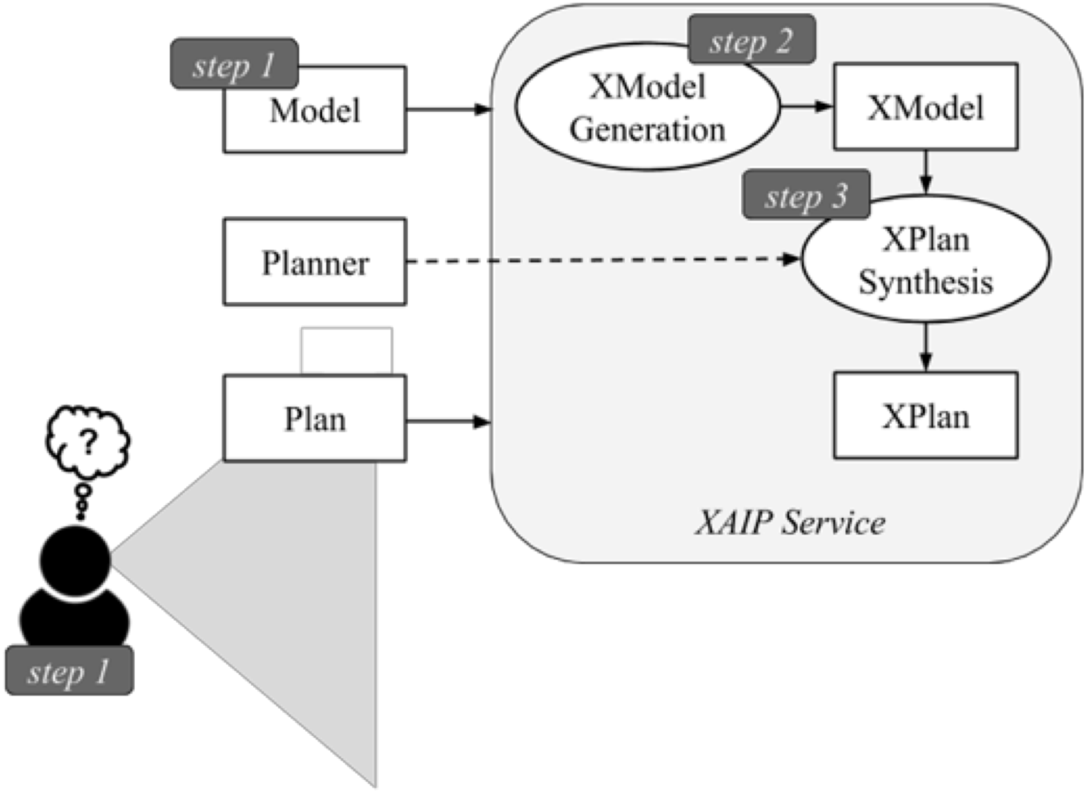
Explainable Planning as a Service

Step 2: Deriving the Explanation Model



Explainable Planning as a Service

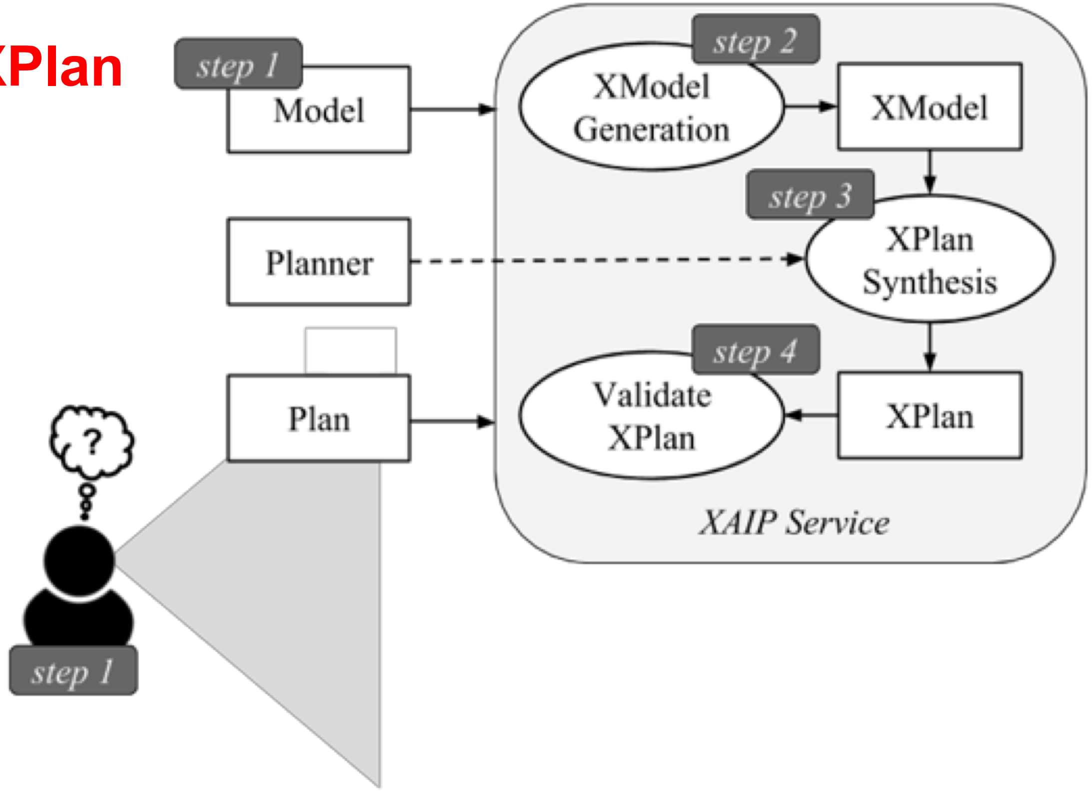
Step 3: Producing the alternative plan (XPlan)



The original planner must be used

Explainable Planning as a Service

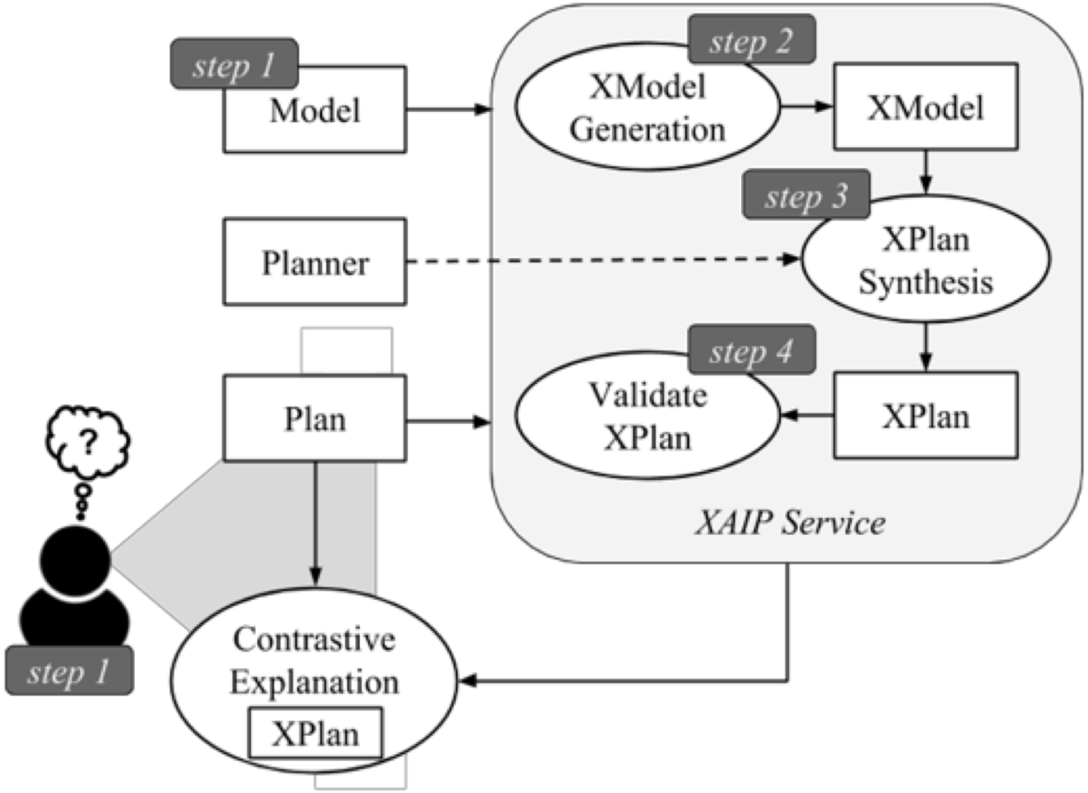
Step 4: Validation of the XPlan



The original planner must be used
The XPlan must be VALid according to the original model

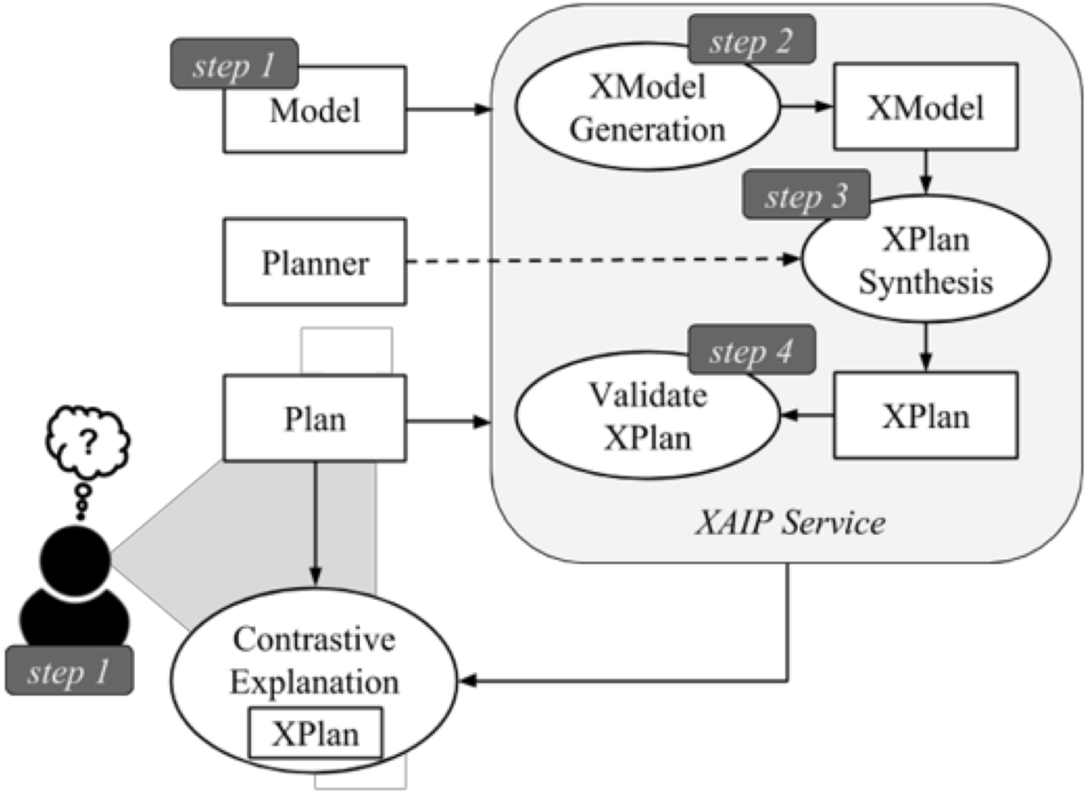
Explainable Planning as a Service

Forming the Contrastive Explanation



Explainable Planning as a Service

Iterative Process !



Explainable Planning as a Service

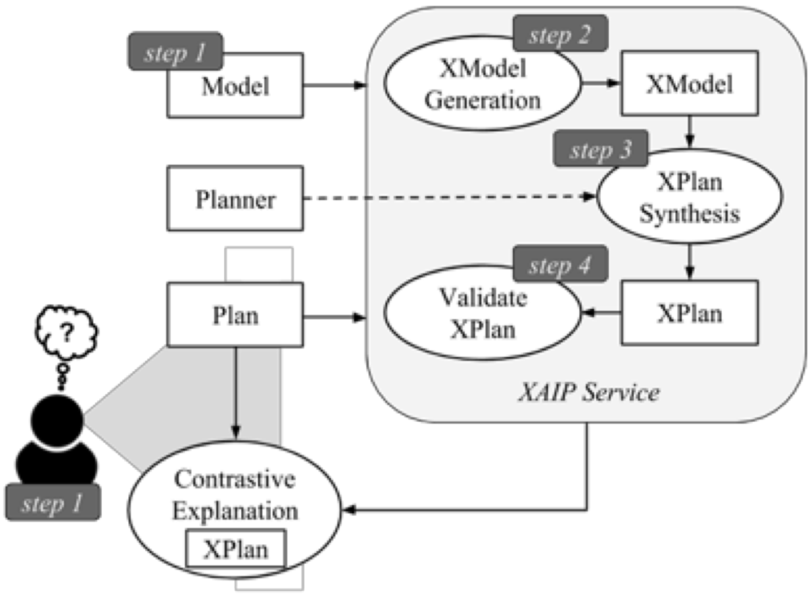
How the user question can be understood, properly taking into account the context in which it was asked?

How to formally characterize the set of questions that can be answered with contrastive explanations?

How constraints can be formally encoded in the XModel?

How to present explanations to the users?

How to assess the effectiveness of the provided explanations?



2nd ICAPS Workshop on Explainable Planning (XAIP-2019)

Berkeley, CA, 11-12 July 2019.

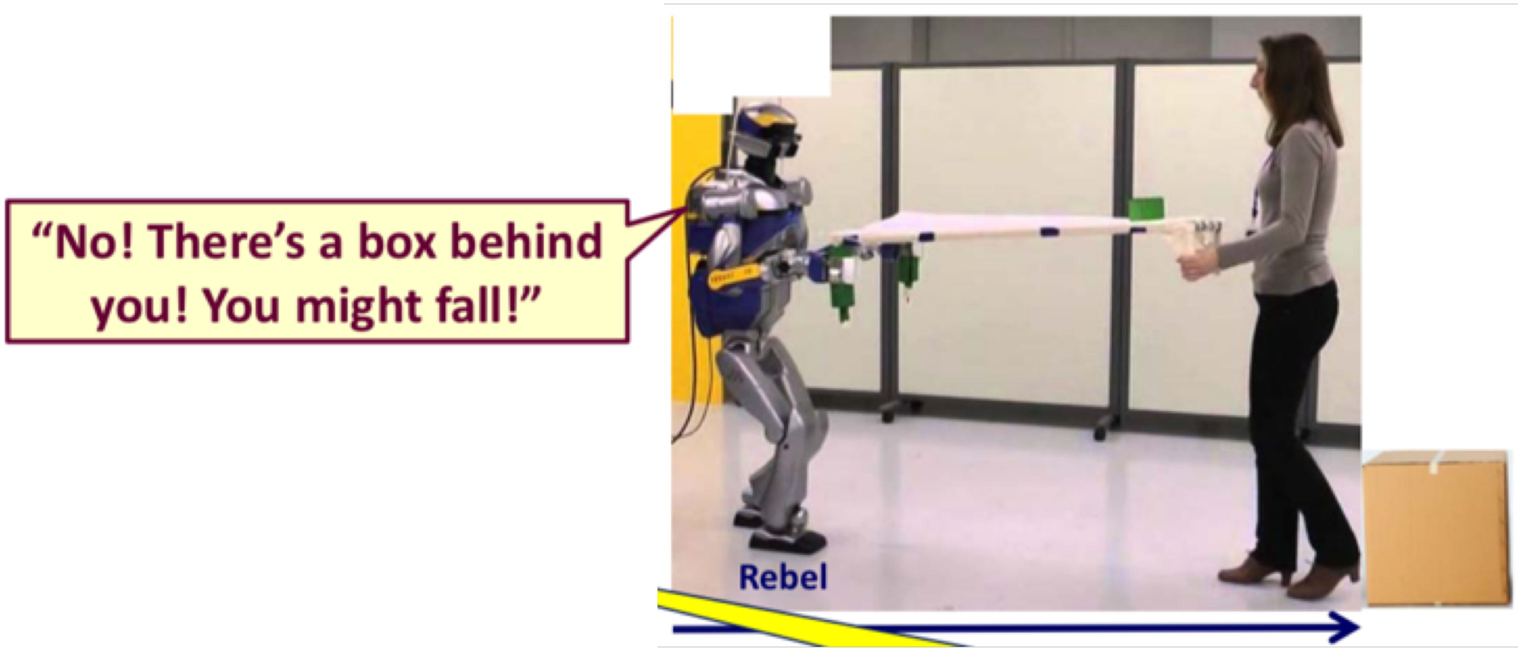
Co-Chairs:
Tathagata Chakraborti (IBM Research AI, USA)
Dustin Dannenhauer (Naval Research Laboratory, USA)
Joerg Hoffmann (Saarland University, Germany)
Daniele Magazzeni (King's College London, UK)

Submission deadline: 22 March 2019

Explaining Rebel Behavior in Goal Reasoning Agents.

D. Dannenhauer, M. Floyd, D. Magazzeni, D. Aha.

Proceedings of ICAPS-18 Workshop on Explainable Planning.



Explainable Security

Luca Viganò and **Daniele Magazzeni**
Department of Informatics
King's College London

Who gives the explanation?

- Designer and/or developer
- User and/or client
- Attacker(s)
- Analyst and/or defender

Who receives the explanation?

- Designer and/or developer
- User and/or client
- Attacker(s)
- Analyst and/or defender

How to explain security?

- Natural language (structured text)
- Graphical languages (trees, graphs, patterns, diagrams, …)
- Formal languages (proofs, plans)
- Gamification (conscious vs unconscious learning)

Why do we need XSec?

- All stakeholders want security
- Confidence, trust, transparency, usability, usage, accountability, verifiability, testability

What is explained?

- Desiderata of the client
- How to use system securely
- System specification and/or code
- Vulnerability and/or attack
- System model
- Security properties
- Threat model

Where is the explanation?

- As part of security/privacy policy
- Detached from system
- Explanation-carrying-system
- Explanation-as-a-service

When is the explanation given?

- Design
- Installation
- Analysis
- Defense
- Implementation
- Use
- Attack
- Modification

Explainable Security has unique and complex characteristics:

- it involves **several different stakeholders** (developers, analysts, users and attackers) and
- is **multi-faceted by nature** (it requires reasoning about system model, threat model, properties of security, privacy and trust, concrete attacks, vulnerabilities, countermeasures).

Who?	What?	Where?
• All stakeholders might need explanations or need to act as explainer	• Explain several "things", at different levels of detail and with different aims	• Explanations can be made available in different places (X-carrying most promising)
When?	**Why?**	**How?**
• Design time • Runtime • Post-hoc	• We make too many mistakes because we don't understand or don't explain	• Explain proof or attack? • Explain explanation process • Trade-off with security threats

If you explain too much, they will attack:

- Explanations might provide information that an attacker can exploit.
- Explanations might need to be "relativized" and made less "powerful" by withholding details.

Trust in Human-Machine Partnership (THuMP)

KING'S College LONDON

THuMP focusses on planning and allocation of resources in critical domains, bringing together experts in AI, Law and Social Science.

1. What are the technical challenges involved in creating Explainable AI Planning systems?
2. What are the technical, legal and social challenges involved in instantiating with explanations a planning system for solving resource allocation problems in critical domains?
3. What are the legal and social implications of enhancing machines with transparency and the ability to explain?

Schlumberger Save the Children. Hogan Lovells EPSRC

Model-Based Explainable AI for Safe and Trusted Human-Autonomy Teaming

Daniele Magazzeni

KING'S College LONDON

@DanMagazzeni