Formal Modelling and Analysis of a Self-Adaptive Robotic System

or On Modelling and Analysing Autonomous Underwater Robots as Probabilistic Featured Transition Systems

Juliane Päßler, Maurice H. ter Beek, Ferruccio Damiani,
S. Lizeth Tapia Tarifa, Einar Broch Johnsen

T-LADIES 1st year meeting

13 October 2023

link to paper

REMARO
RELIABLE AI FOR MARINE ROBOTICS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956200

Problem

Analyse
Self-Adaptive Systems
as
Dynamic Software
Product Lines
with
ProFeat

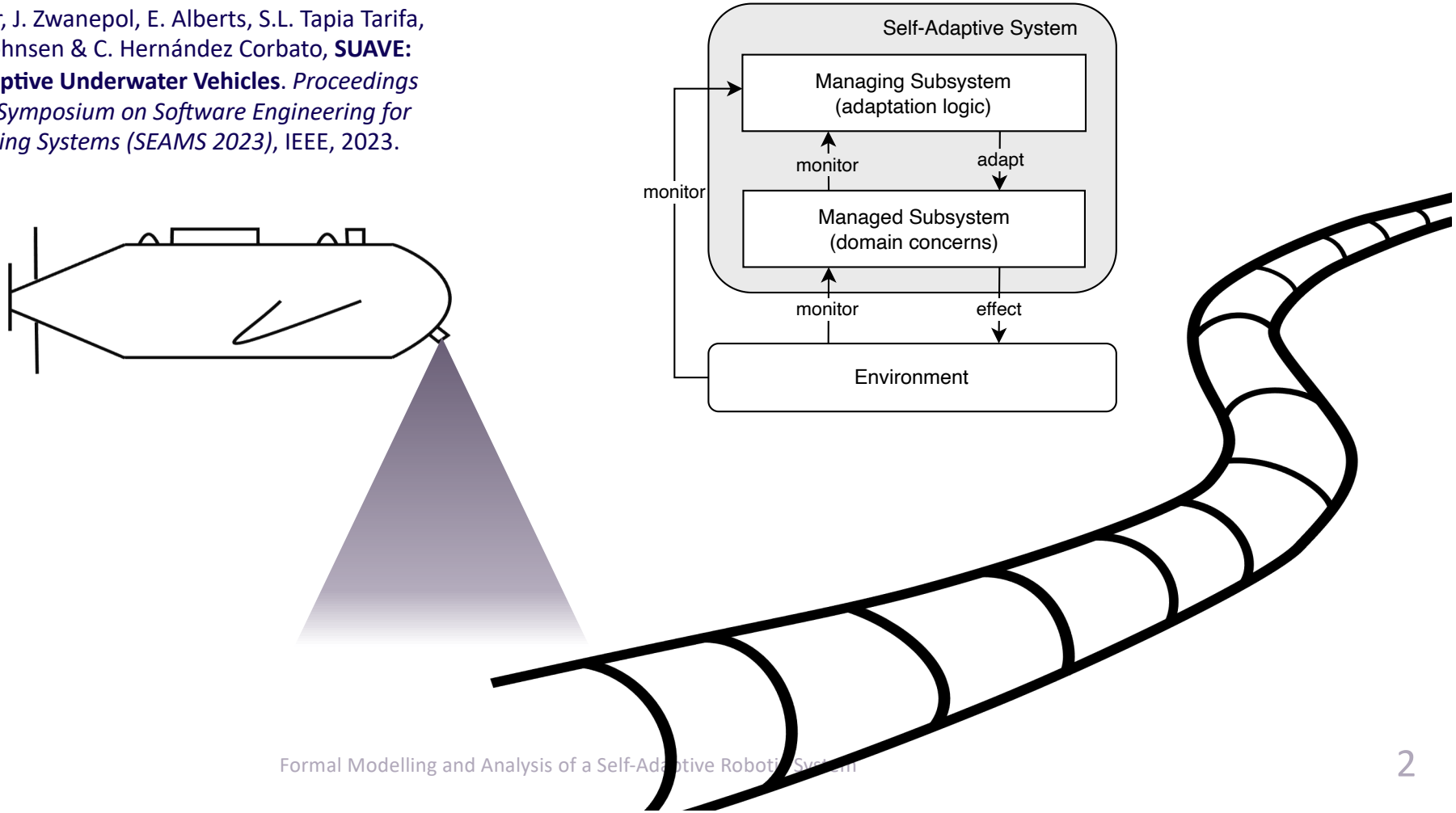
P. Chrszon, C. Dubslaff, S. Klüppelholz, C. Baier,
ProFeat: Feature-Oriented Engineering for Family-Based Probabilistic Model Checking.
Formal Aspects of Computing 30 (2018), 45–75.

https://pchrszon.github.io/profeat

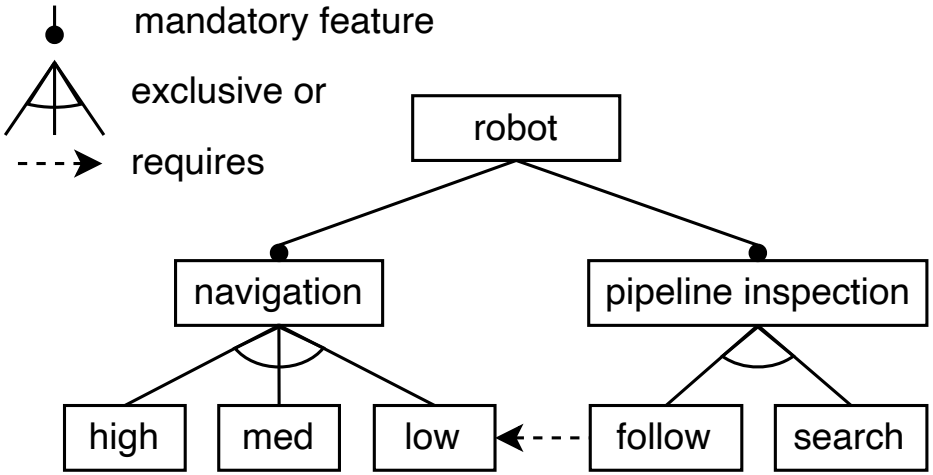
Pipeline Inspection

Formal Methods and Tools Lab

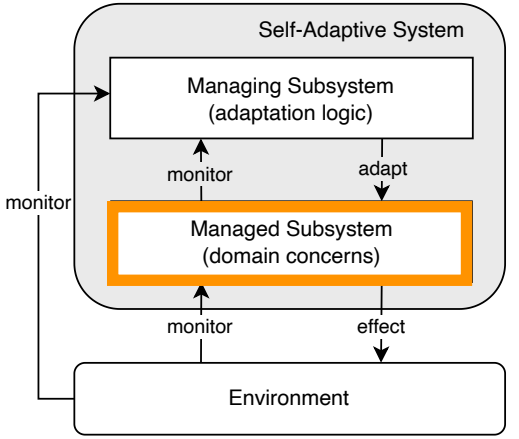
G. Rezende Silva, J. Päßler, J. Zwanepol, E. Alberts, S.L. Tapia Tarifa, Gerostathopoulos, E.B. Johnsen & C. Hernández Corbato, **SUAVE: An Exemplar for Self-Adaptive Underwater Vehicles**. *Proceedings of the 18th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2023)*, IEEE, 2023.

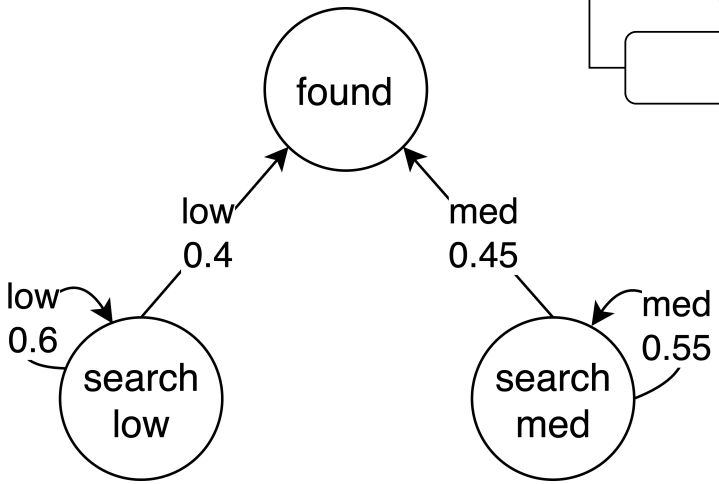


Probabilistic Featured Transition System



Feature Model

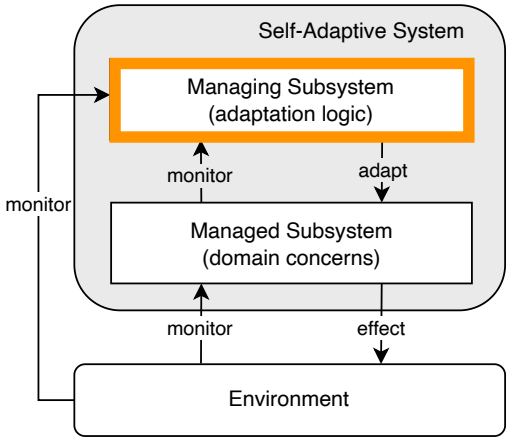




Probabilistic FTS

A. Classen, M. Cordy, P.-Y. Schobbens, P. Heymans, A. Legay, J.-F. Raskin, **FTS: Foundations for Verifying Variability-Intensive Systems and Their Application to LTL Model Checking**. *IEEE Transactions on Software Engineering* 39 (2013)

Managing Subsystem



mandatory feature

exclusive or

requires

robot

navigation

pipeline inspection

high

med

low

follow

search

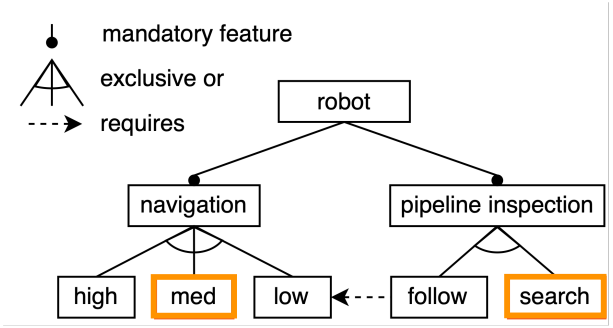
water visibility
medium

searching

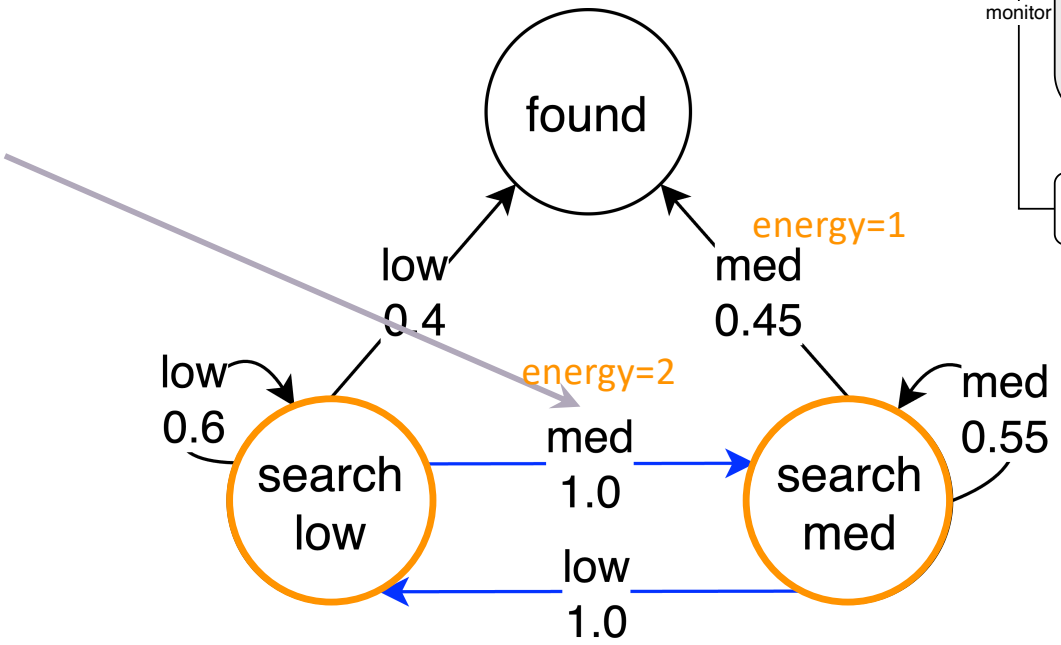
Environment

Managed Subsystem

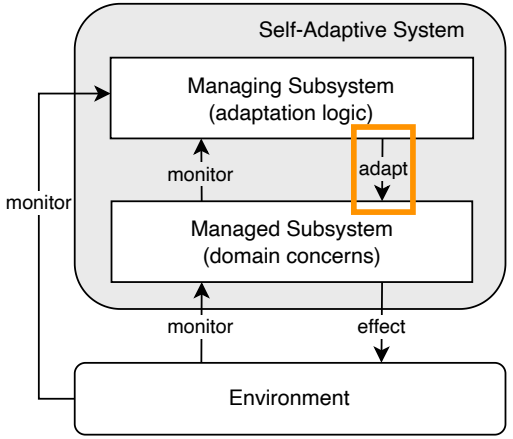
Adaptation



Managing Subsystem



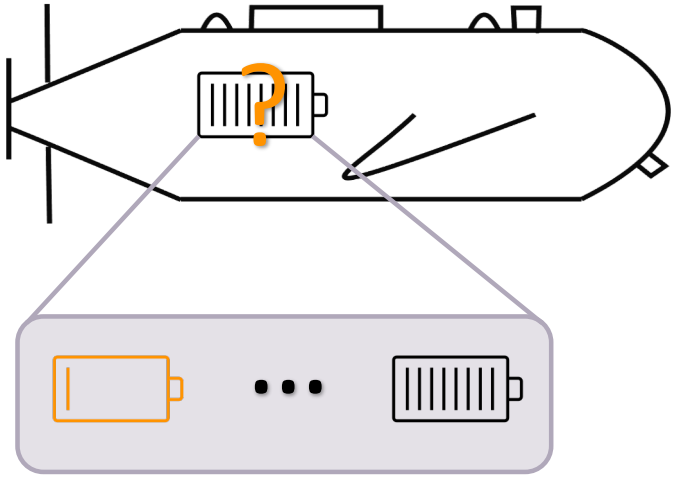
Managed Subsystem



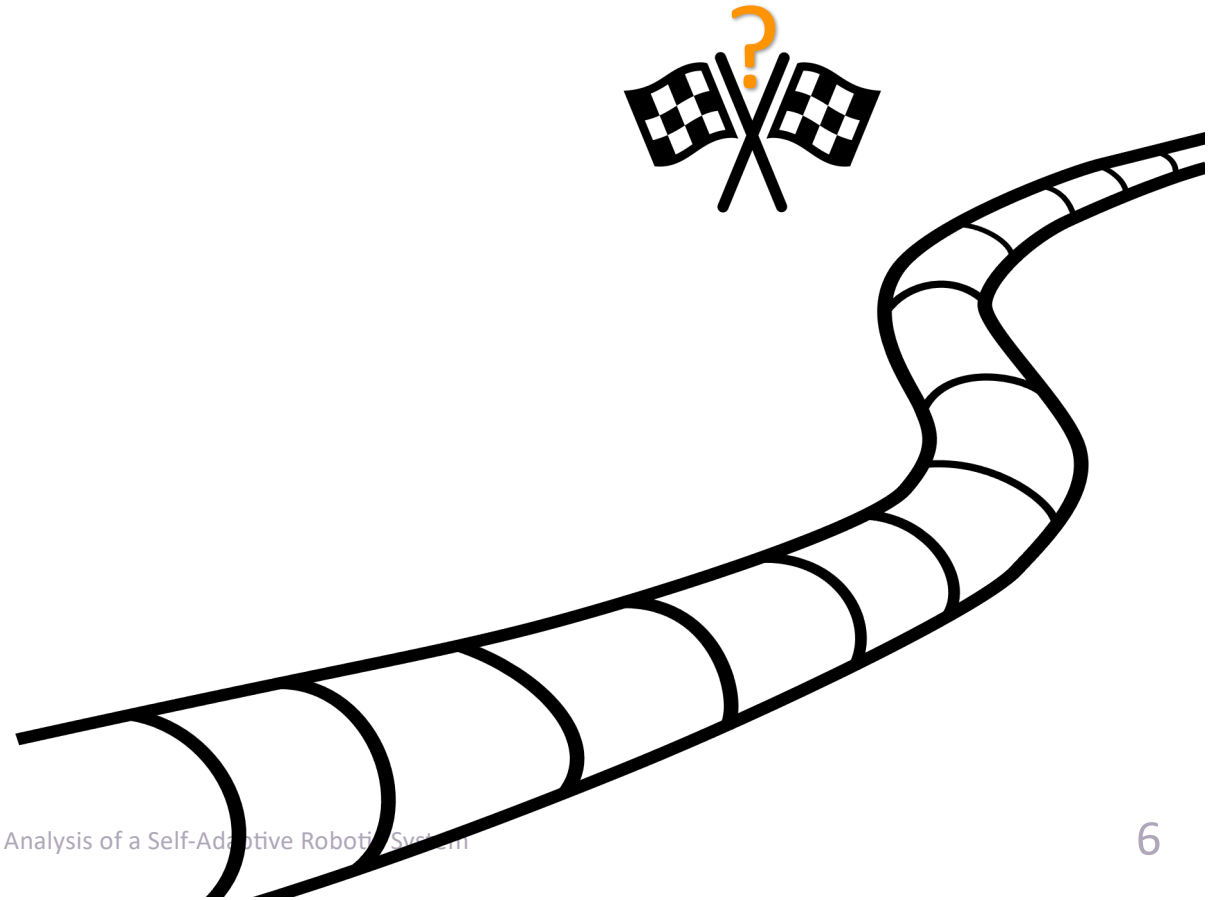
Analysis

Formal Methods and Tools Lab

Costs/Rewards



Probabilities



ProFeat Model of AUV

Formal Methods and Tools Lab

Costs/Rewards

```
root feature
  all of robot;
  modules auv, environment;
  rewards "time"
    [step] true : 1;
  endrewards
  rewards "energy"
    // Costs for being in a recovery state
    (s=recover_high) : 2;
    // .. omitted code ..

    // Costs for switching altitudes
    (s=search_high) & active(low) : 4;
    (s=search_high) & active(med) : 2;
    (s=found) & active(high) : 4;
    (s=found) & active(med) : 2;
    // .. omitted code ..
  endrewards
endfeature
```

Probabilities

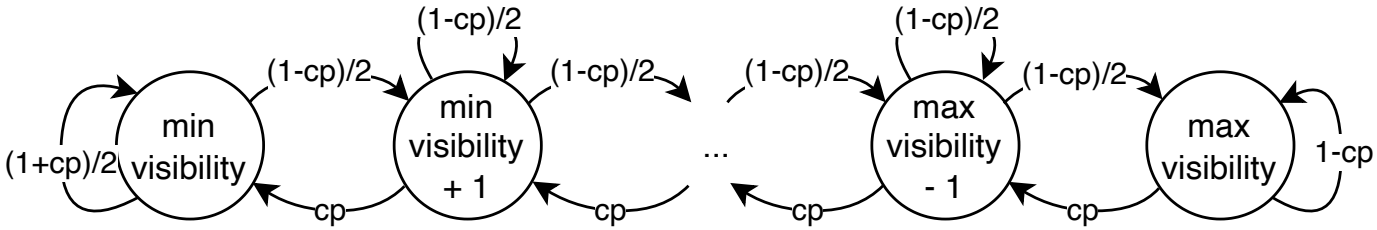
```
module auv
  s : [0..12] init start_task;
  d_insp : [0..inspect] init 0;
  t_failed : [0..infl_tf] init 0;

  // .. omitted code ..
  // From search state to another state
  [step] (s=search_high & active(high)) -> 0.59:(s'=found)
            + 0.4:(s'=search_high) + 0.01:(s'=recover_high);
  [step] (s=search_high & active(med)) -> 1:(s'=search_med);
  [step] (s=search_high & active(low)) -> 1:(s'=search_low);
  // .. omitted code ..

  // Following the pipeline
  [step] (s=following) & (d_insp<inspect) & (t_failed=0)
          -> 0.9: (s'=following) & (d_insp'=d_insp+1)
          + 0.07: (s'=lost_pipe) + 0.03:(s'=recover_following)
          & (t_failed'=(t_failed<infl_tf? t_failed+1 : t_failed));
  // .. omitted code ..
  [step] (s=following) & (d_insp=inspect) -> (s'=done);

  // Lost the pipeline
  [step] (s=lost_pipe) -> 1: (s'=start_task) & (t_failed'=0);
  // .. omitted code ..
endmodule
```

ProFeat Model of Environment




```
module environment
  water_visib : [min_visib..max_visib]
           init round((max_visib-min_visib)/2);
  [step] true -> current_prob: (water_visib'= (water_visib=min_visib?
           min_visib:water_visib-1)) + (1-current_prob)/2: (water_visib'=
           (water_visib=max_visib? max_visib:water_visib+1))
          + (1-current_prob)/2: true;
endmodule
```

ProFeat Feature Controller

Formal Methods and Tools Lab

```
1  formula med_visib = (max_visib-min_visib)/3;
2  formula high_visib = 2*(max_visib-min_visib)/3;
3
4  controller
5    // Change altitude depending on water visibility
6    [step] (s!=found) & active(search) & water_visib < med_visib
7           -> activate(low) & deactivate(high) & deactivate(med);
8    [step] (s!=found) & active(search)
9                & med_visib <= water_visib & water_visib < high_visib
10          -> activate(low) & deactivate(med) & deactivate(high);
11   [step] (s!=found) & active(search)
12               & med_visib <= water_visib & water_visib < high_visib
13          -> activate(med) & deactivate(low) & deactivate(high);
14   // .. omitted code ..
15
16   // Switch task from "search" to "follow"
17   [step] (s=found) & active(search)
18          -> deactivate(search) & activate(follow) & activate(low)
19               & deactivate(med) & deactivate(high);
20
21   // Switch task from "follow" to "search"
22   [step] (s=lost_pipe) & active(follow)
23          -> deactivate(follow) & activate(search);
24
25   // Enable transitions when following the pipeline
26   [step] (s!=lost_pipe) & active(follow) -> true;
27 endcontroller
```

ProFeat Analysis: Energy and Duration

Formal Methods and Tools Lab

Scenario	min_visib	max_visib	current_prob	inspect
1 (North Sea)	1	10	0.6	10
2 (Caribbean Sea)	3	20	0.3	30

```
1 R{"energy"}min=? [F ${s=done}];
2 R{"energy"}max=? [F ${s=done}];
```

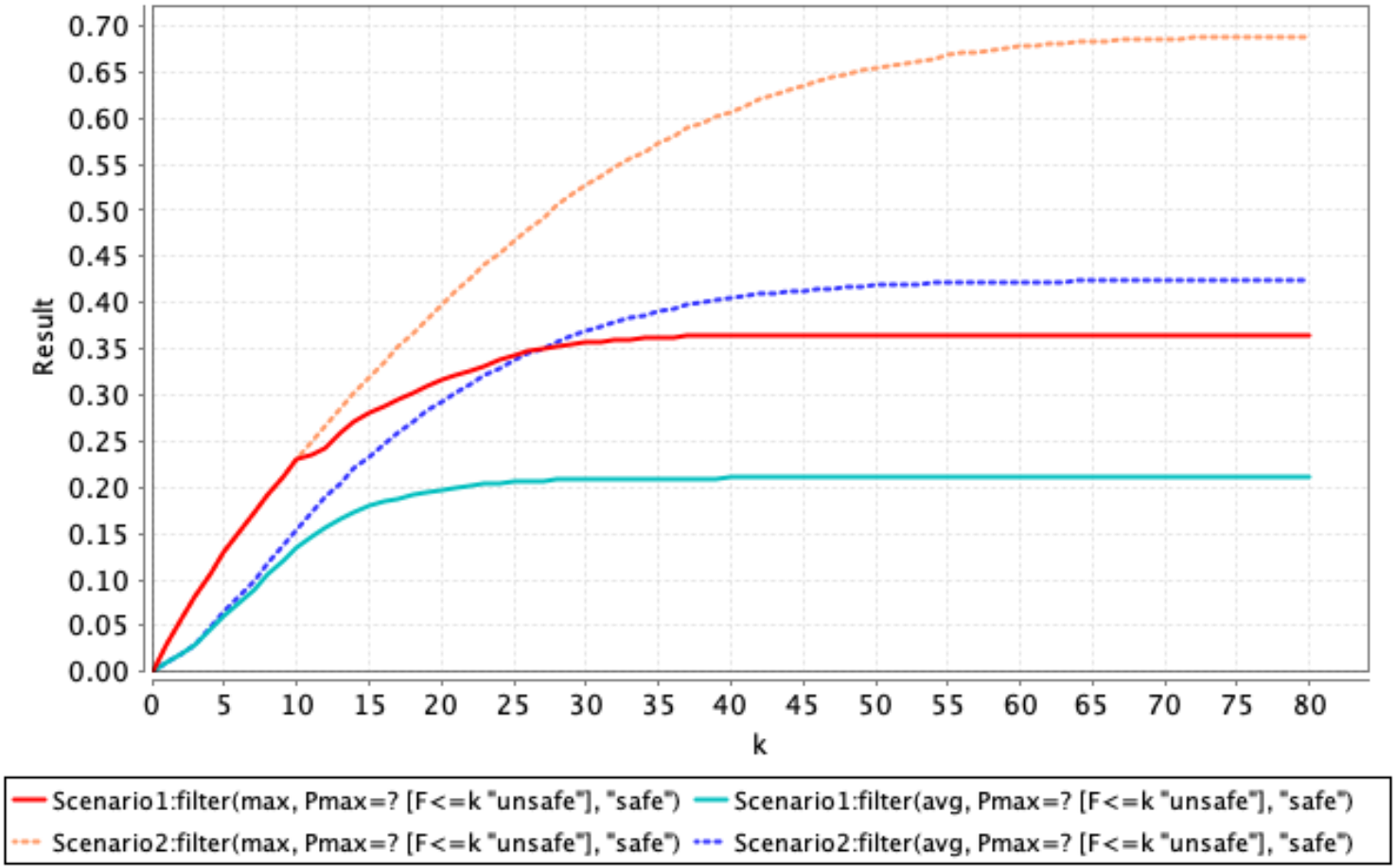
	Energy		Time	
Scenario	min	max	min	max
1	24.78	44.39	23.66	32.40
2	59.08	4723.29	55.54	1315.58

ProFeat Analysis: Unsafe States

Formal Methods and Tools Lab

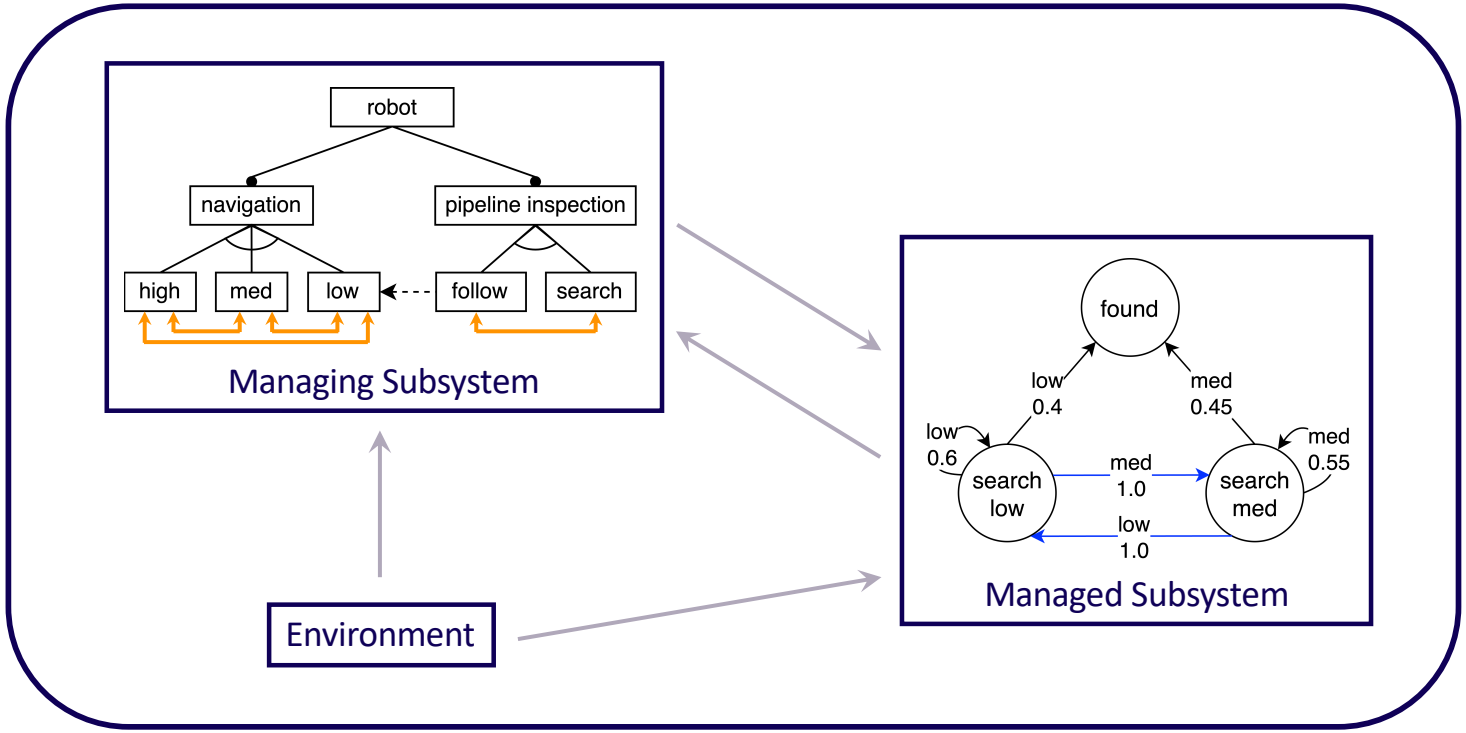
```
label "unsafe" = s=recover_high | s=recover_med | s=recover_low
               | s=recover_following;
label "safe" = s=start_task | s=lost_pipe | s=start_search | s=search_high
               | s=search_med | s=search_low | s=found | s=following |s=done;
Pmin=? [G "safe"];
filter(min, Pmin=? [ F<=k "safe" ], "unsafe");
filter(max, Pmax=? [ F<=k "unsafe" ], "safe");
filter(avg, Pmax=? [ F<=k "unsafe" ], "safe");
```

ProFeat Analysis: Unsafe States



Summary

Analyse SASs as DSPLs: ProFeat with family-based analysis



link to paper

REMARO
RELIABLE AI FOR MARINE ROBOTICS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956200