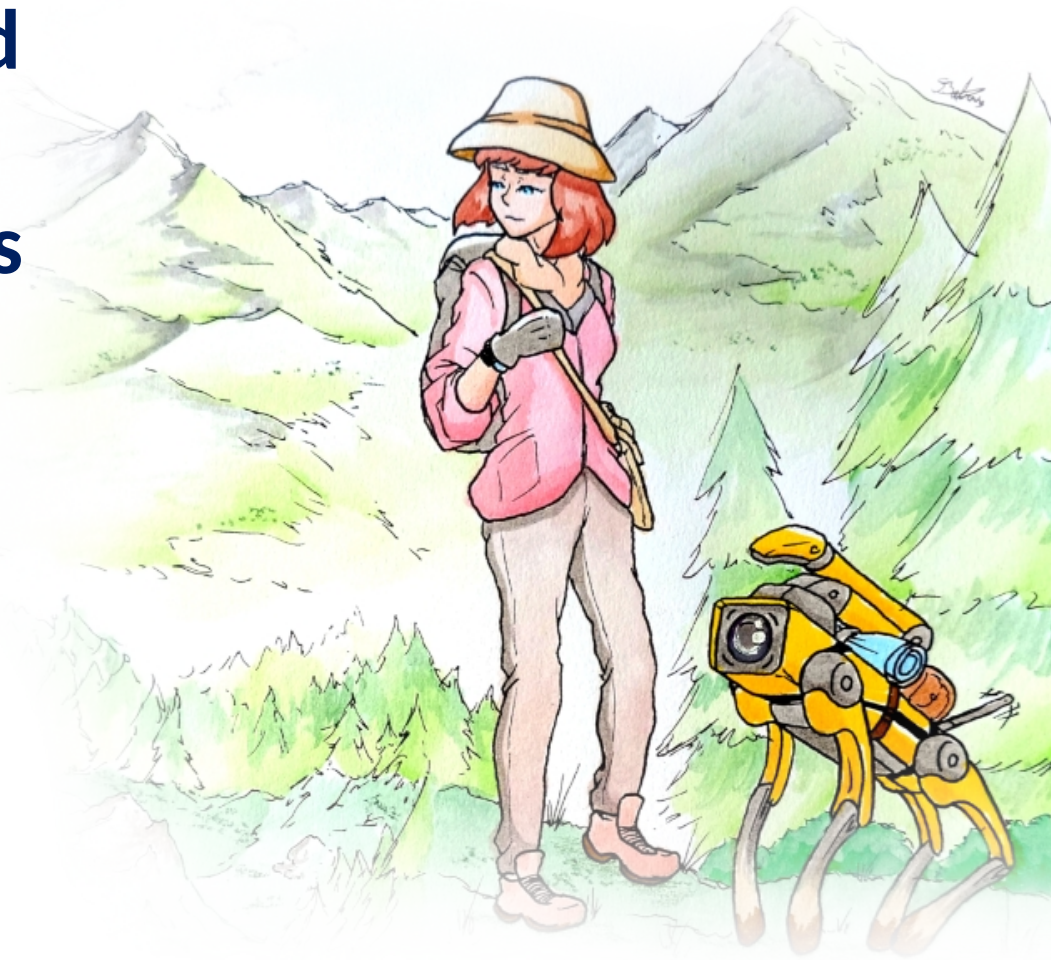
SkiNet : a user-oriented tool for PN-based analysis of robotic skills



Baptiste Pelletier (ISAE)
Charles Lesire (ONERA)
Karen Godary-Dejean (LIRMM)

Petri Nets 25 - 26/06/2025

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

anr agence nationale de la recherche
AU SERVICE DE LA SCIENCE

isae
Institut Supérieur de l'Aéronautique et de l'Espace
SUPAERO

ONERA
THE FRENCH AEROSPACE LAB

La Région Occitanie
Pyrénées - Méditerranée

LIRMM

Content

1. Context
2. SkiNet overview
3. User-space VS PN-space
4. Live demo

Content

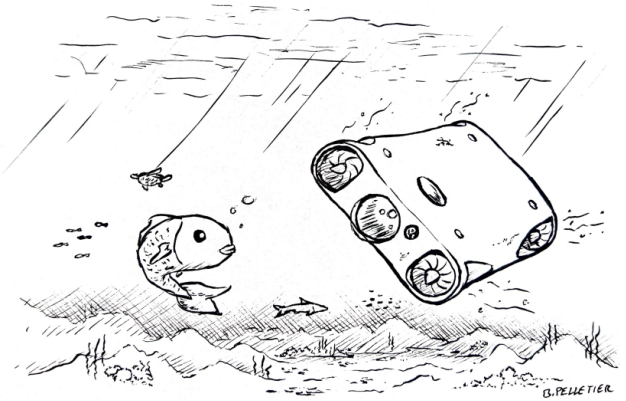
1. Context
2. SkiNet overview
3. User-space VS PN-space
4. Live demo

Context : FMs in robotics

Bring trust to autonomous systems – Focus on robotics

Motivations

- Autonomy increases the cost and complexity
- Task and environments get more complex
- Human help may be difficult or impossible (underwater, military, space…)



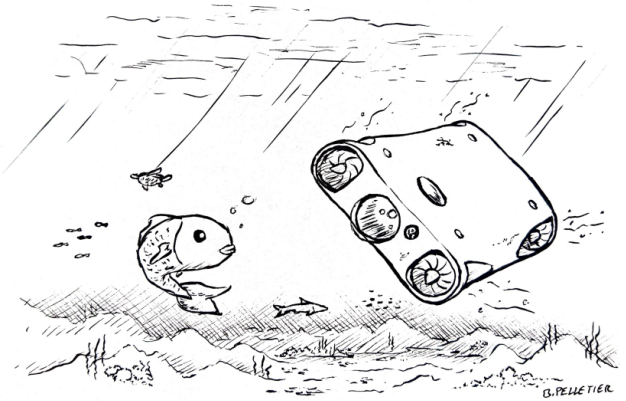
Dangerous job for South African wildlife rangers despite new technology (Michael Strang/UN Environment-GEF rhino project, Aug 2016)

Context : FMs in robotics

Bring trust to autonomous systems – Focus on robotics

Motivations

- Autonomy increases the cost and complexity
- Task and environments get more complex
- Human help may be difficult or impossible (underwater, military, space…)



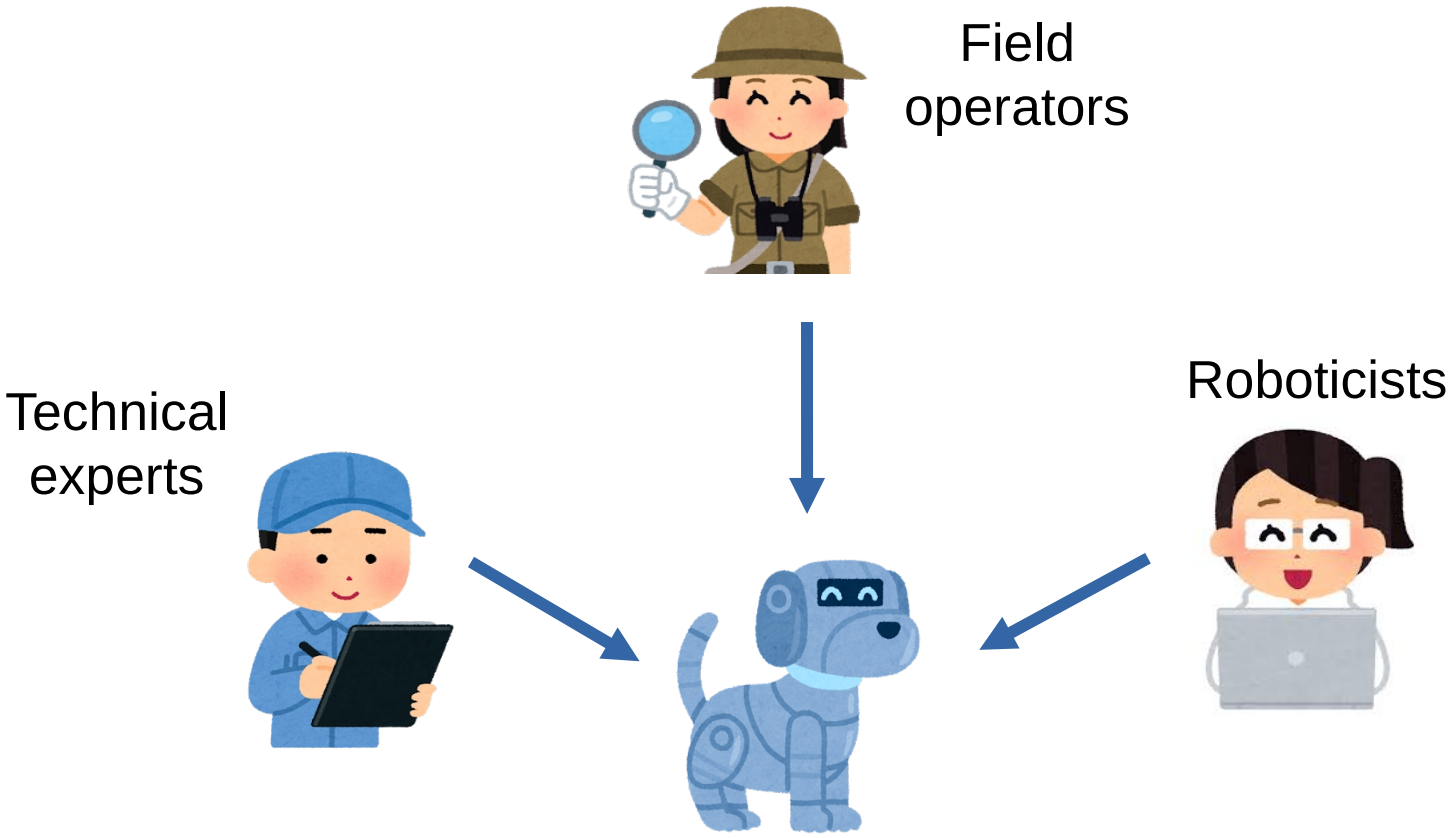
Apply formal methods to bring trust to the system

- Prove the feasability of a mission
- Improve fault-tolerance
- Anticipate runtime problems

Dangerous job for South African wildlife rangers despite new technology (Michael Strang/UN Environment-GEF rhino project, Aug 2016)

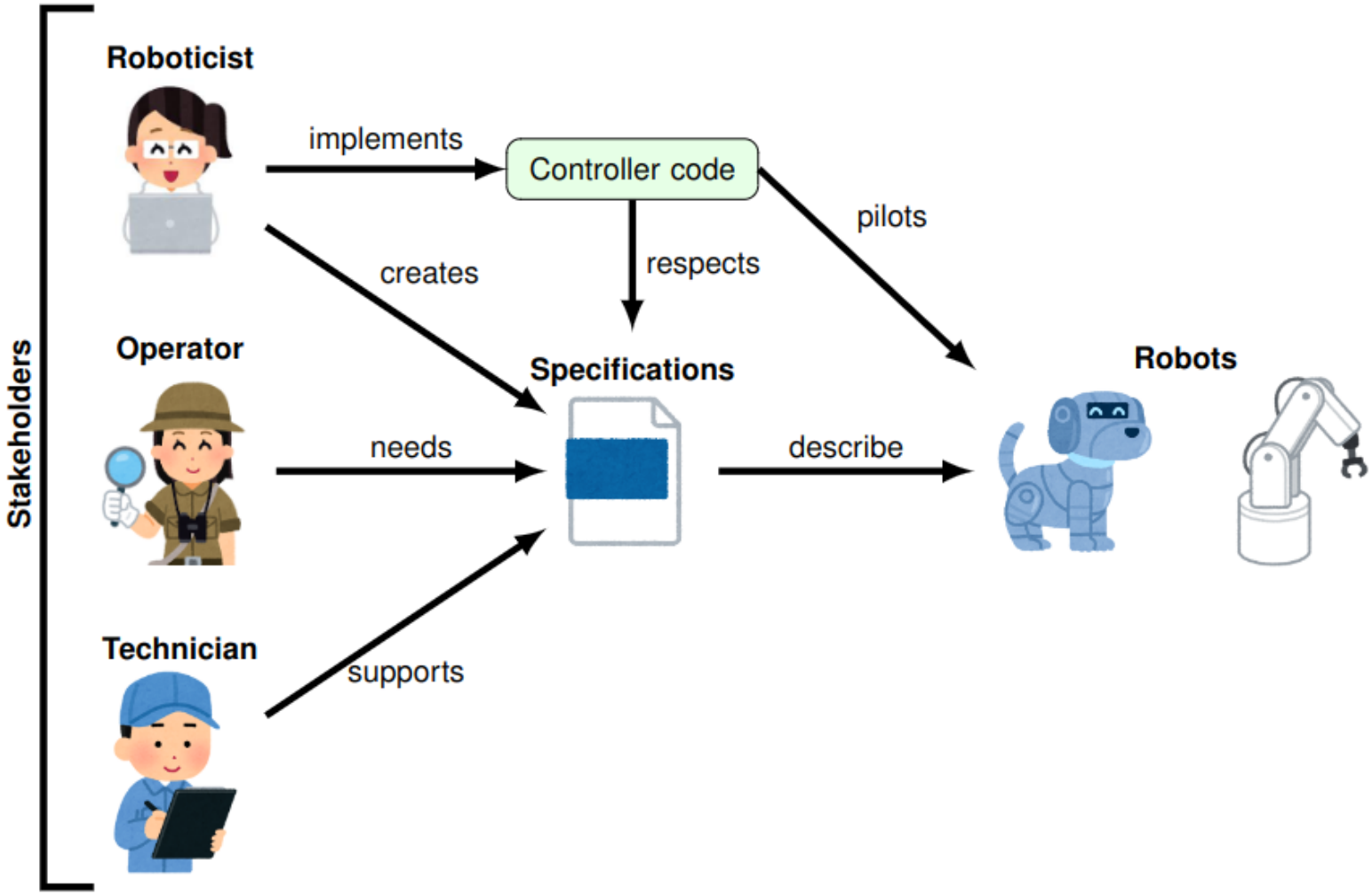
Context : FMs in robotics

Problem: actors have heterogenous backgrounds…



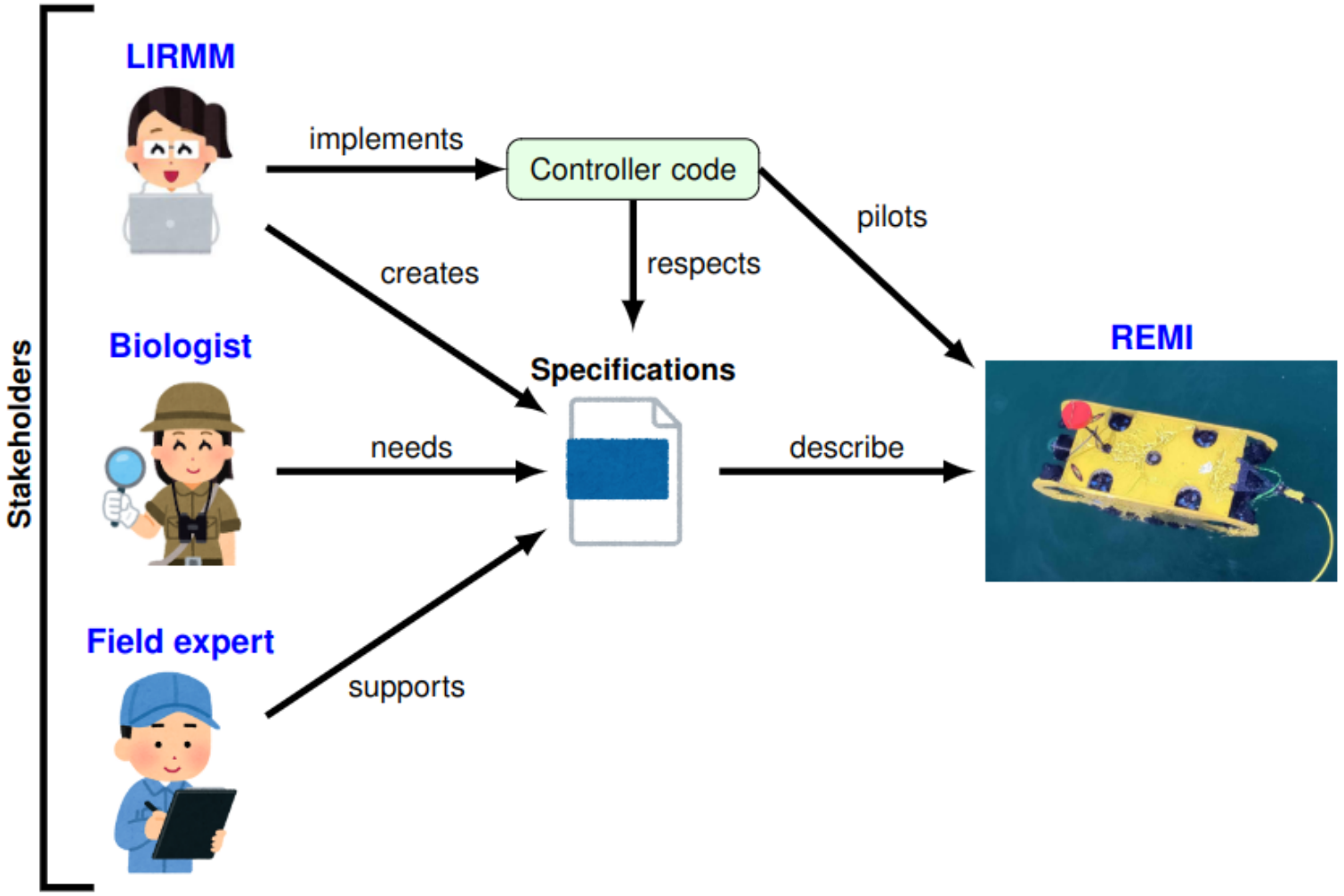
Context : FMs in robotics

Problem: actors have heterogenous backgrounds...



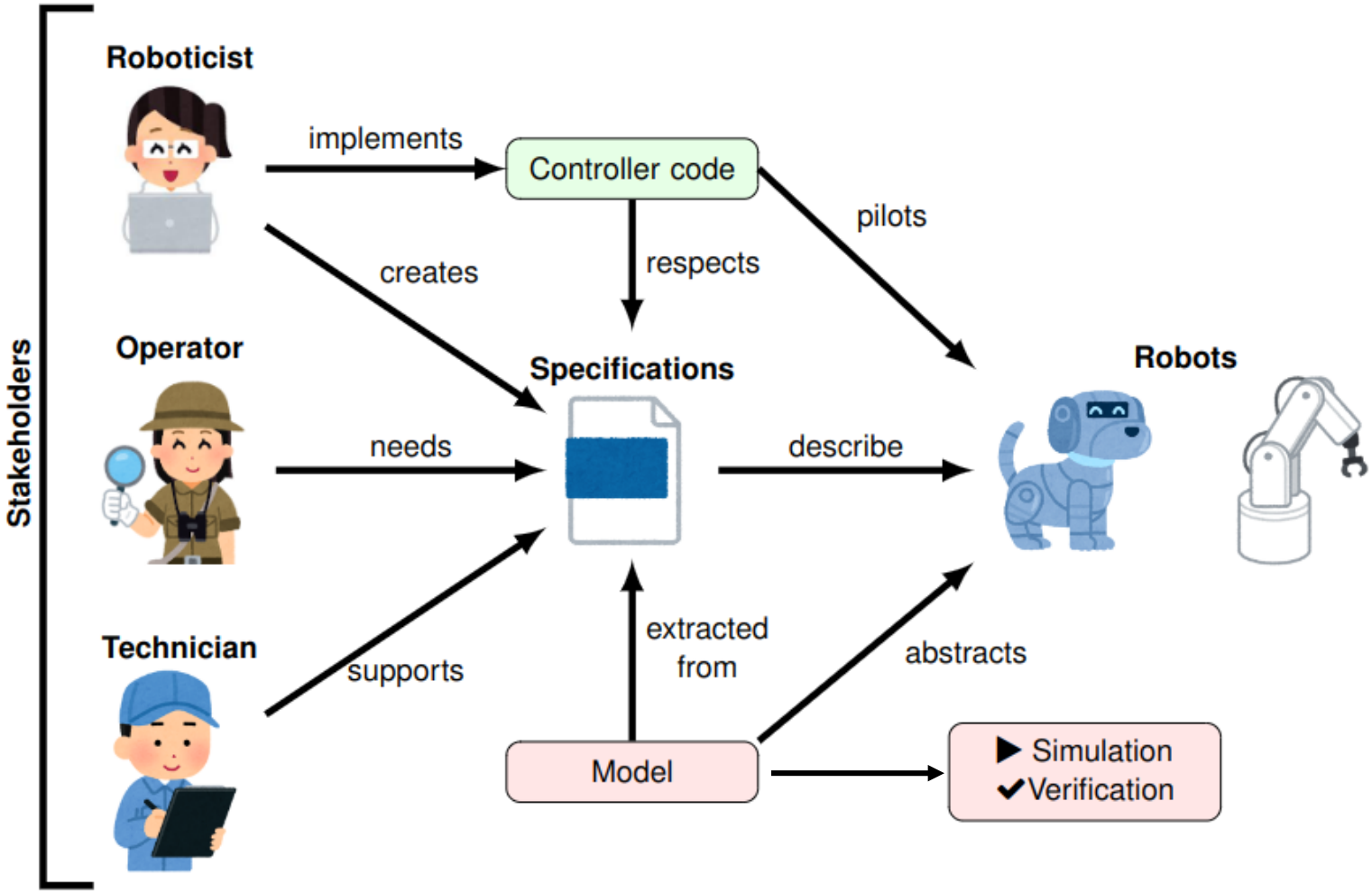
Context : FMs in robotics

Problem: actors have heterogenous backgrounds...



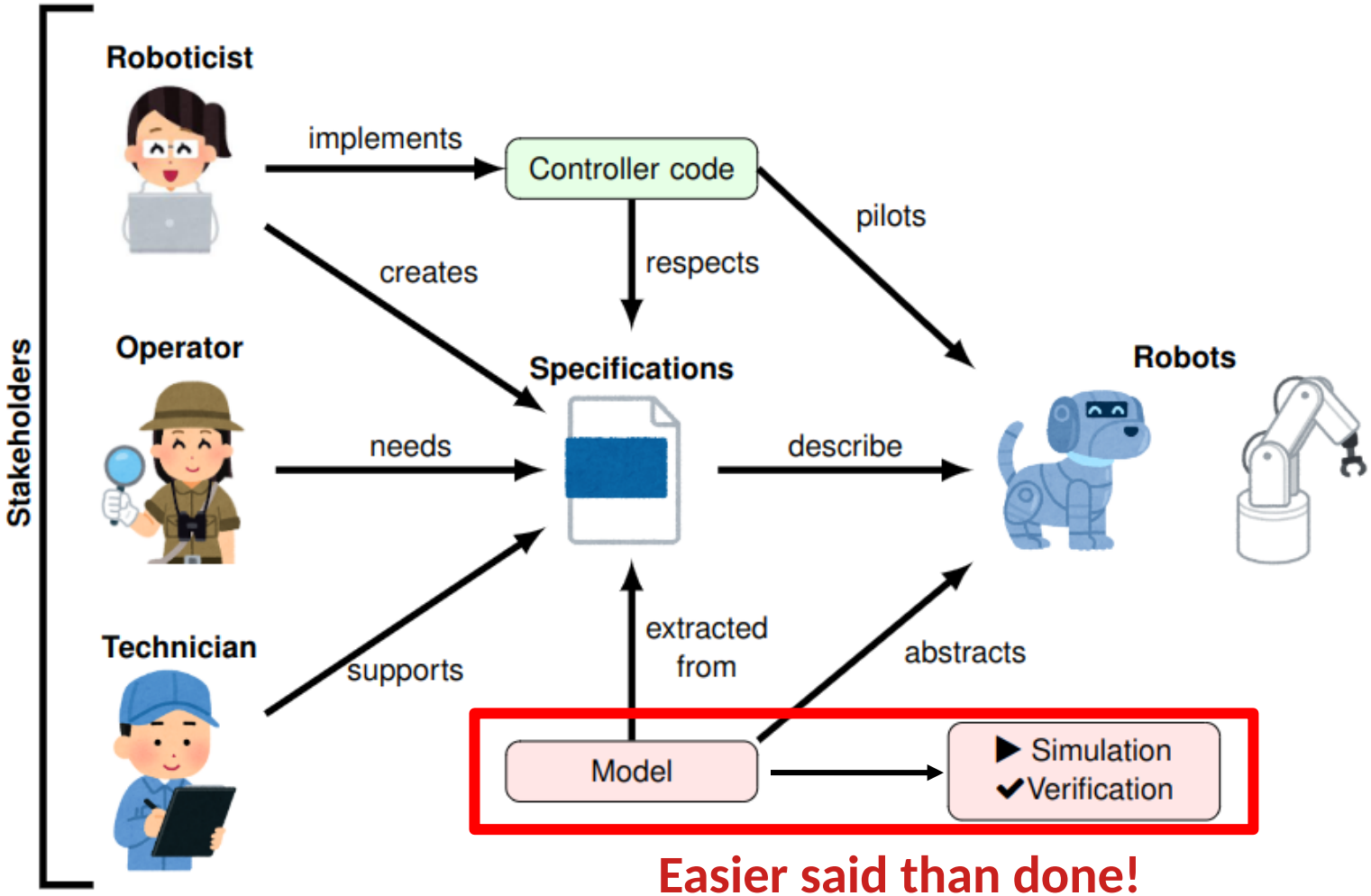
Context : FMs in robotics

Problem: actors have heterogenous backgrounds...



Context : FMs in robotics

Problem: actors have heterogenous backgrounds...



Easier said than done!

Context : FMs in robotics

Problem: actors have heterogenous backgrounds… But who will get their hands dirty?

What's a robot?

AF (x=5 => y>2)

I can't keep up!

What's a Petri net?

Context : FMs in robotics

Problem: actors have heterogenous backgrounds… But who will get their hands dirty?

The perfect specimen

- Has never heard of FMs
- 3 hours of experience in robotics
- Full of hope and optimism

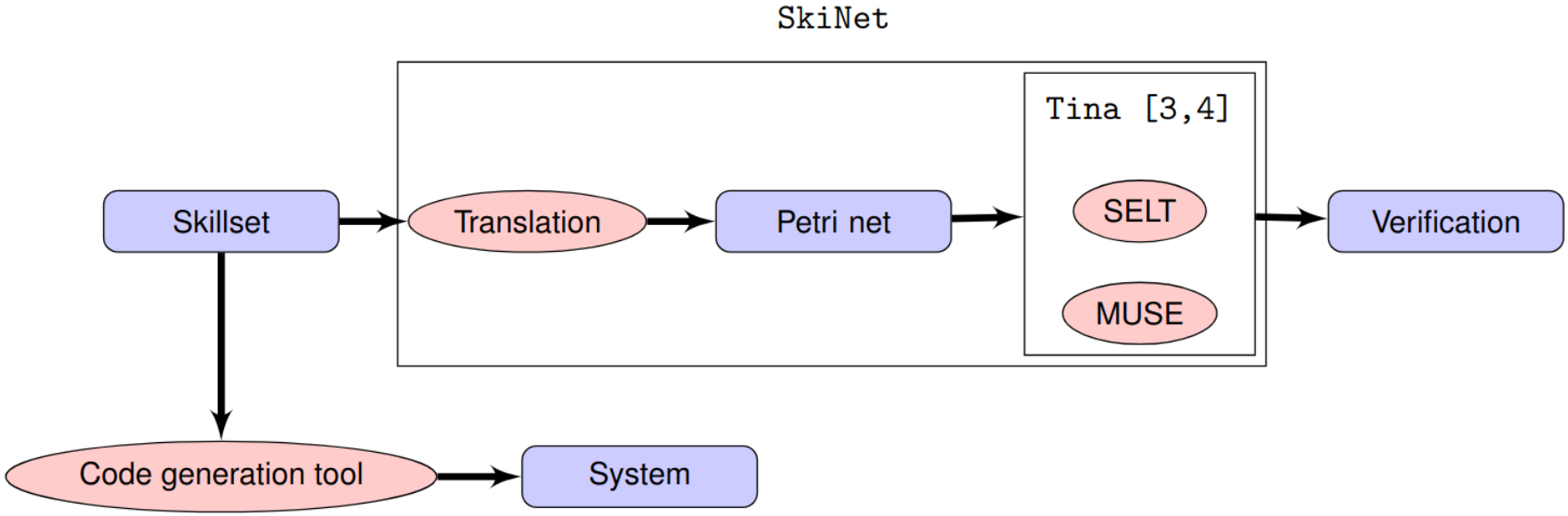
Content

1. Context
2. SkiNet overview
3. User-space VS PN-space
4. Live demo

SkiNet Overview

A transformer from Robotic Skills Language specifications [Albore et al., 2023] to PNs

Pelletier et al., FMAS (formal methods for autonomous systems), 2022



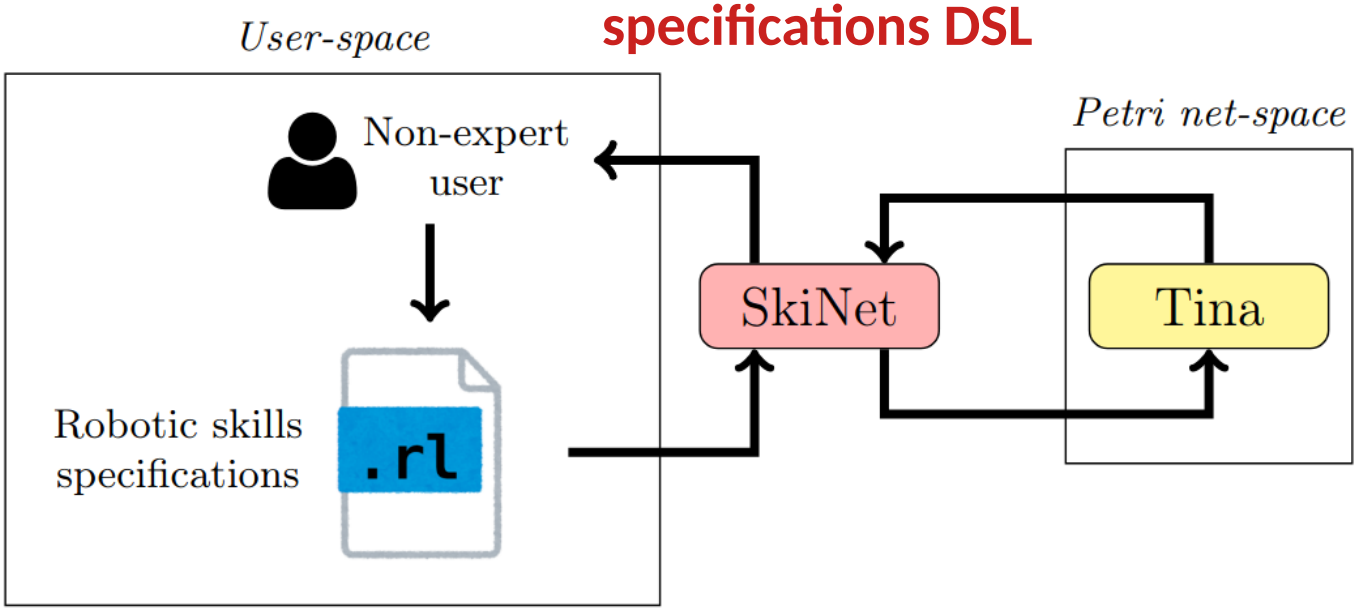
[3] B. Berthomieu, P.-O. Ribet and F. Vernadat (2004): The tool TINA – Construction of abstract state spaces for petri nets and time petri nets. International Journal of Production Research 42, pp. 2741–2756, doi:10.1080/00207540412331312688.
[4] Bernard Berthomieu, François Vernadat and Silvano dal Zilio (2004): The TINA toolbox Home Page - Time Petri Net Analyzer - by LAAS/CNRS. Available at https://projects.laas.fr/tina/home.php

SkiNet Overview

Now a back-and-forth transformer!

Pelletier et al., 2025 (this paper)

The user evolves the specifications DSL



The tool serves as an interface to use FMs

Content

1. Context
2. SkiNet overview
3. User-space VS PN-space
4. Live demo

User-space: Robotic Skills Language

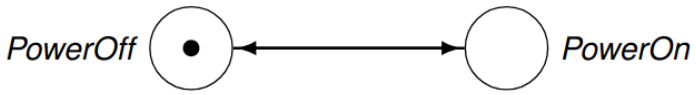
```
skillset spot_fmas {

    resource {
        power_status {
            initial PowerOff
            PowerOff -> PowerOn
            PowerOn -> PowerOff
        }
        lease_status {
            initial AutoMode
            AutoMode -> ManualMode
            ManualMode -> AutoMode
        }
        control_mode {
            initial Idle
            Idle -> Busy
            Busy -> Idle
        }
    }
```

Albore et al., 2023

Resources $\mathcal{R}$ = **finite-state machines** (FMSs)

- **Internal elements**: battery level, stance, power status, control mode, mutex, sensor data...
- **External elements**: operator control, environment objects, mission elements...

Ex. `power_status`:



User-space: Robotic Skills Language

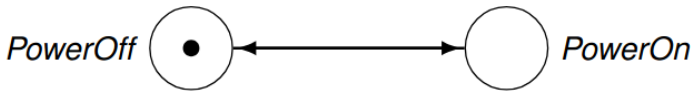
```
skillset spot_fmas {

    resource {
        power_status {
            initial PowerOff
            PowerOff -> PowerOn
            PowerOn -> PowerOff
        }
        lease_status {
            initial AutoMode
            AutoMode -> ManualMode
            ManualMode -> AutoMode
        }
        control_mode {
            initial Idle
            Idle -> Busy
            Busy -> Idle
        }
    }

        event {
            toauto_frommanual {
                guard lease_status == ManualMode
                lease_status -> AutoMode
            }
            tomanual_fromauto {
                guard lease_status == AutoMode
                lease_status -> ManualMode
            }
            power_switchoff {
                guard power_status == PowerOn
                power_status -> PowerOff
            }
            power_switchon {
                guard power_status == PowerOff
                power_status -> PowerOn
            }
        }
```

Albore et al., 2023

Resources $\mathcal{R}$ = **finite-state machines** (FMSs)

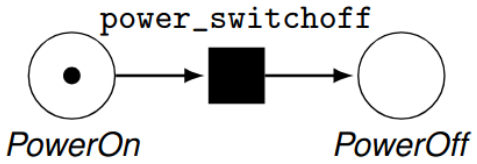
- **Internal elements**: battery level, stance, power status, control mode, mutex, sensor data...
- **External elements**: operator control, environment objects, mission elements...

Ex. `power_status`:



Events $\mathcal{V}$ = **uncontrollable elements** that can affect the state of resources.

- **Resource guard** = **logical formula** $\phi : \{S^r, r \in \mathcal{R}\} \rightarrow \{True, False\}$
- **Effect** $\epsilon = (r, S_i^r)$ of a resource and its **next state**.



User-space: Robotic Skills Language

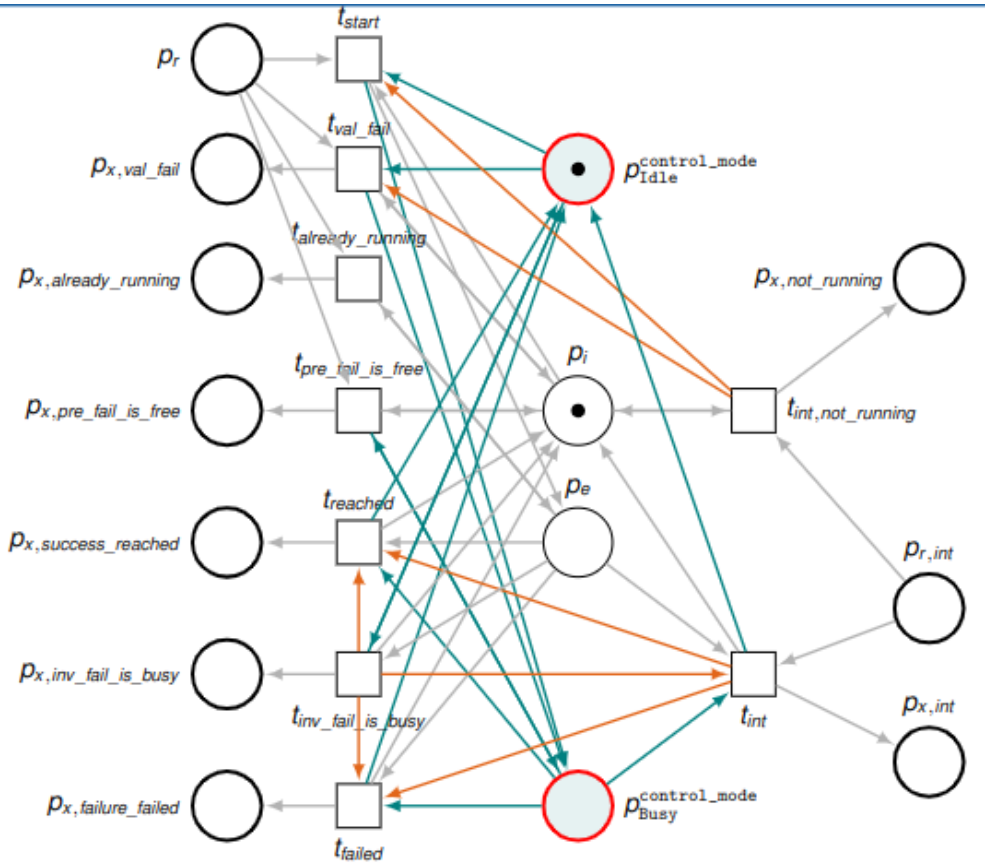
Albore et al., 2023

```
skill go_to_graphnav {
    input {
        waypoint : GeoPoint
    }
    precondition {
        is_free : control_mode == Idle
    }
    start { control_mode -> Busy }
    invariant {
        is_busy {
            guard control_mode == Busy
        }
    }
    interrupt { effect control_mode -> Idle }
    success reached { effect control_mode -> Idle }
    failure failed { effect control_mode -> Idle }
}
```

PN-space: PN model & Tina

Albore et al., 2023

```
skill go_to_graphnav {
    input {
        waypoint : GeoPoint
    }
    precondition {
        is_free : control_mode == Idle
    }
    start { control_mode -> Busy }
    invariant {
        is_busy {
            guard control_mode == Busy
        }
    }
    interrupt { effect control_mode -> Idle }
    success reached { effect control_mode -> Idle }
    failure failed { effect control_mode -> Idle }
}
```



PN-space: PN model & Tina

But models are rarely simple…

- Resources to model hardware/software
- Skills for autonomous actions and user-operated
- Events for each uncontrollable action that could hit the robot (increases when modelling fault tolerance)

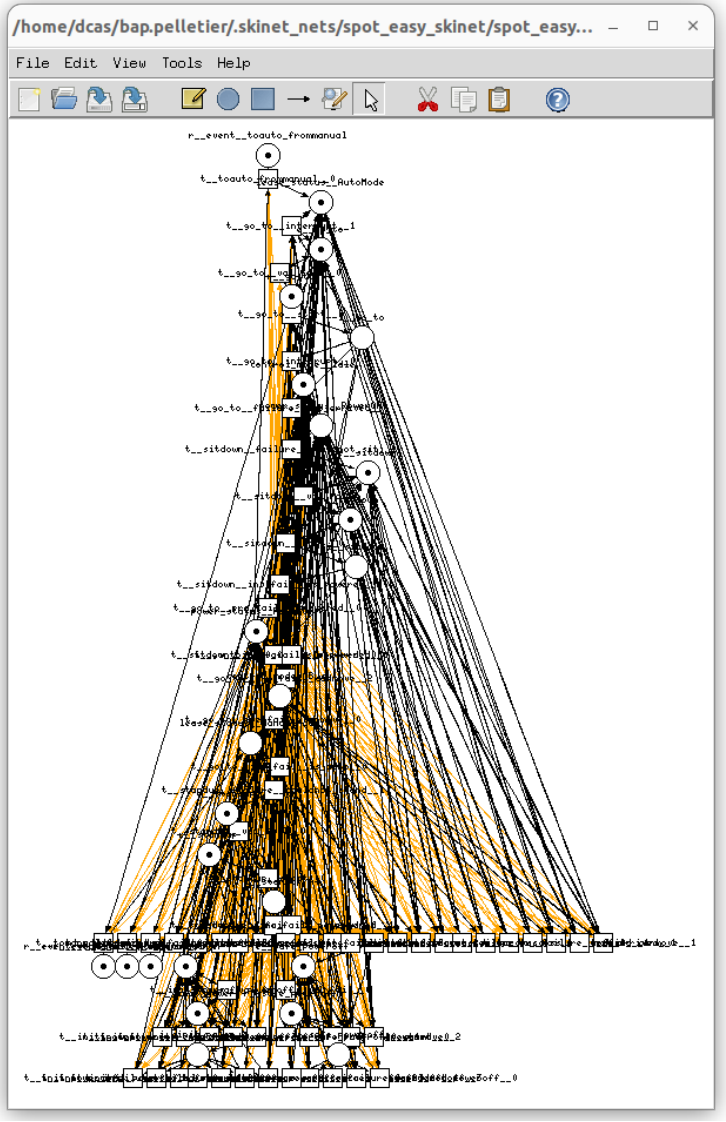
Skillset	Res	Sk	Pl	Tr	States
UGV	3	5	24	60	37
Spot	6	10	48	164	400
UAV	6	10	54	297	1512
Jaco	4	6	49	266	**51340**
Jaco (6DOF)	6	1	63	16089	**16265**

PN-space: PN model & Tina

But models are rarely simple…

- Resources to model hardware/software
- Skills for autonomous actions and user-operated
- Events for each uncontrollable action that could hit the robot (increases when modelling fault tolerance)

Skillset	Res	Sk	Pl	Tr	States
UGV	3	5	24	60	37
Spot	6	10	48	164	400
UAV	6	10	54	297	1512
Jaco	4	6	49	266	**51340**
Jaco (6DOF)	6	1	63	16089	**16265**



PN-space: PN model & Tina

But models are rarely simple…

- Resources to model hardware/software
- Skills for autonomous actions and user-operated
- Events for each uncontrollable action that could hit the robot (increases when modelling fault tolerance)

Skillset	Res	Sk	Pl	Tr	States
UGV	3	5	24	60	37
Spot	6	10	48	164	400
UAV	6	10	54	297	1512
Jaco	4	6	49	266	**51340**
Jaco (6DOF)	6	1	63	16089	**16265**

Graphical simulation/debugging is impossible

PN-space: PN model & Tina

```
Play version 3.7.5 -- 03/29/23 -- LAAS/CNRS
parsed net spot_easy
0.004s

initial
state 0: control_mode__Idle e__go_to e__init_power e__safe_poweroff e__sitdown e__
standup lease_status__AutoMode power_status__PowerOff r__event__power_switchoff r_
_event__power_switchon r__event__toauto_frommanual r__event__tomanual_fromauto r__
go_to r__init_power r__safe_poweroff r__sitdown r__standup
enabled: t__go_to__pre_fail__ispowered__0 t__go_to__val_fail__0 t__init_power__sta
rt__0 t__init_power__val_fail__0 t__power_switchon__0 t__safe_poweroff__pre_fail__
ispowered__0 t__safe_poweroff__val_fail__0 t__sitdown__pre_fail__ispowered__0 t__s
itdown__val_fail__0 t__standup__pre_fail__ispowered__0 t__standup__val_fail__0 t__
tomanual_fromauto__0
firable: t__go_to__pre_fail__ispowered__0 t__go_to__val_fail__0 t__init_power__sta
rt__0 t__init_power__val_fail__0 t__power_switchon__0 t__safe_poweroff__pre_fail__
ispowered__0 t__safe_poweroff__val_fail__0 t__sitdown__pre_fail__ispowered__0 t__s
itdown__val_fail__0 t__standup__pre_fail__ispowered__0 t__standup__val_fail__0 t__
tomanual_fromauto__0

?
```

Marking

Firable transitions

Enabled transitions

Hum...

PN-space: PN model & Tina

Present the simulation as the user expects!

```
RESOURCES                    | SKILLS                 | EVENTS
                             |                        |
power_status        PowerOff | init_power       IDLE  | toauto_frommanual   -----
lease_status        AutoMode | safe_poweroff    PRE!  | tomanual_fromauto   READY
control_mode            Idle | standup          PRE!  | power_switchoff     -----
                             | sitdown          PRE!  | power_switchon      READY
                             | go_to            PRE!  |

 nb enabled: 12
 firable:
   0 - trigger tomanual_fromauto
   1 - trigger power_switchon
   2 - start init_power

- safe_poweroff  : precondition ispowered not met.
- standup        : precondition ispowered not met.
- sitdown        : precondition ispowered not met.
- go_to          : precondition ispowered not met.

?
```

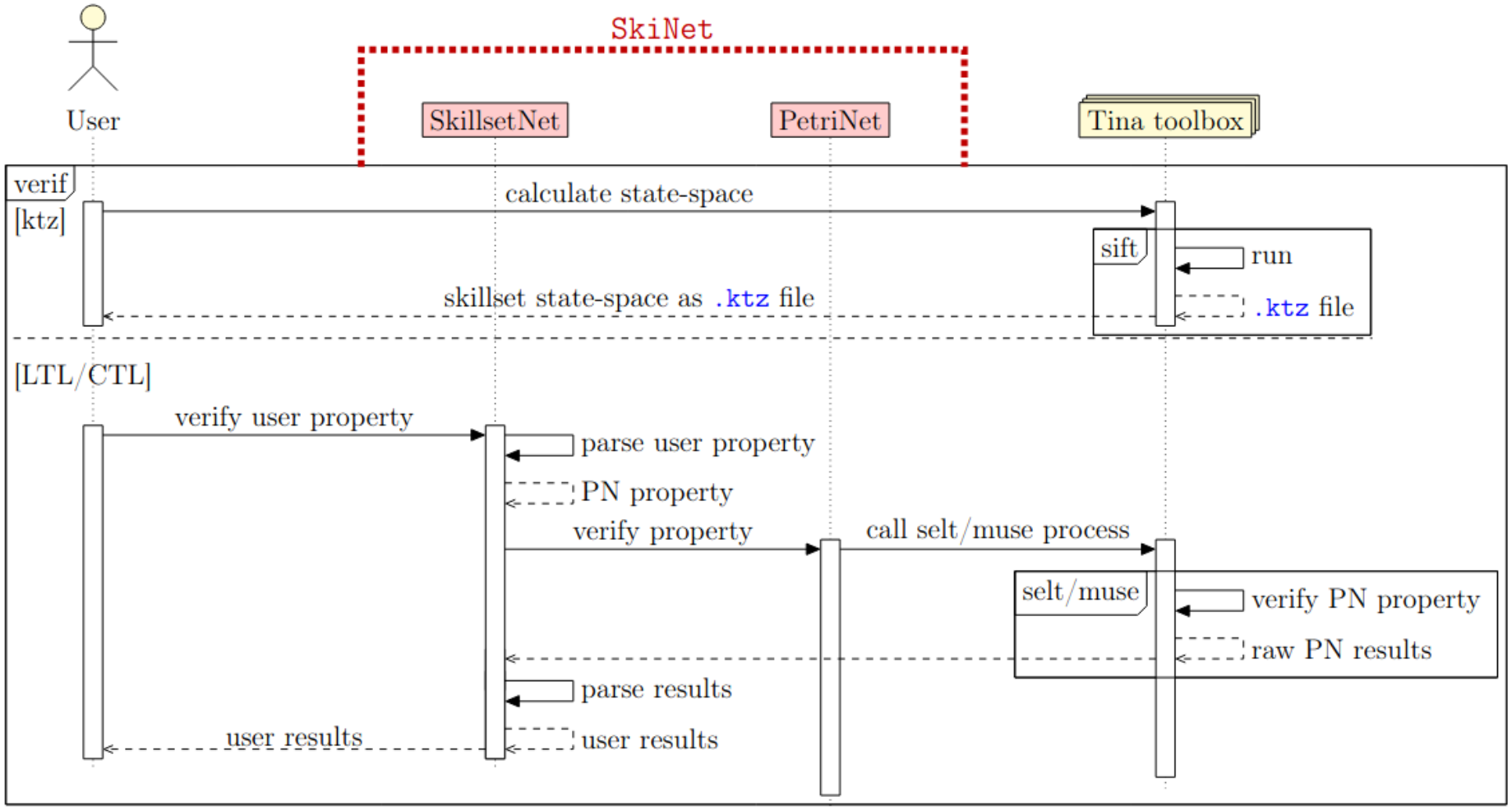
Skillset state

Firable transitions

Additional information

PN-space: PN model & Tina

What about model-checking?



Content

1. Context
2. SkiNet overview
3. User-space VS PN-space
4. Live demo

Conclusion & future works

Goals achieved

- SkiNet turned into a **two-way transformer** and we hope it inspires others to do so!
- Currently **used by students** as part of the embedded software and autonomous systems master courses at ISAE Supaero (Toulouse, FR)

Conclusion & future works

Goals achieved

- SkiNet turned into a **two-way transformer** and we hope it inspires others to do so!
- Currently **used by students** as part of the embedded software and autonomous systems master courses at ISAE Supaero (Toulouse, FR)

Perspectives

- Conduct a **small user-experience survey** on the whole Robotic Skills toolchain (skills DSL, SkiNet, controller code implementation…)
- **Improve the formal models** of skills by adding data to the models
- Improve **modularity** with other formal frameworks (Fiacre, Automata/UPPAAL, BPMN, SMT, …)

Thank you for your attention !

Gitlab : gitlab.com/onera-robot-skills/skinet-release

Pelletier et al., 2022: arxiv.org/pdf/2209.14039