RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

ONERA
THE FRENCH AEROSPACE LAB

Decision-Making and Skill-based architectures for Autonomous Mobile Robots

Alexandre Albore

ONERA, DTIS, Université de Toulouse, France

14 May 2024, Reasoning and Learning for Autonomous Agents: Agents Vic Autumn Symposium

Context

- Ever-increasing robotic applications **demand**
- Robotic approaches to **autonomy**
- Need of **dependability**
- Need for **tools and methods**

Trust and dependability

Strengthening **trust** in autonomous systems

- Complex robotic systems
- Complex tasks and difficult environments
- Exogenous events (HMI, uncertainty, partial information)

Dependability

- **Specification** made easy
- **Verification and validation** made easy
- **Implementation** made easy

A Robotic 3-layered architecture

The proposed architecture has three layers:

1. The **decision layer**: hosts the mission controller and sets the goals to achieve
2. The **executive layer**: the middleware interface of the system capabilities
3. The **functional layer**: the on-board control procedures

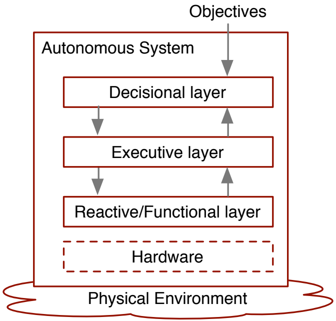
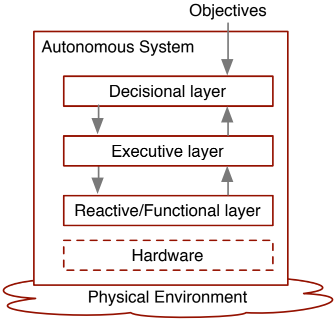


Fig: 3-layers architecture[1]

[1] F. Ingrand and M. Ghallab. "Deliberation for autonomous robots: A survey". In: *Artificial Intelligence* 247 (2017), pp. 10–44

Dependable Skill-based executive layer

- Executive layer:
  - translation of decision *actions* into **executable** commands
  - abstraction of the functional layer
  - “task achieving behaviors”[2], “procedures”[3], “performance profiles”[4], “skills”[5,6,7]
  - **Skills** represent functions/services available in the functional layer



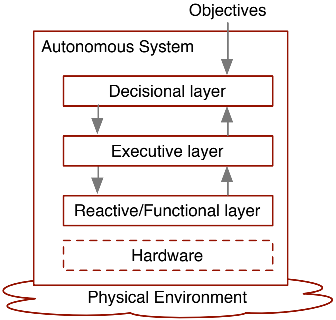
[6] Rodney Brooks. “A robust layered control system for a mobile robot”. In: *IEEE Journal on Robotics and Automation* 2.1 (1986), pp. 14–23.
[7] Felix Ingrand et al. “PRS: A high Level Supervision and Control Language for Autonomous Mobile Robots”. In: *ICRA*. Minneapolis, MN, USA, 1996.
[8] Ronen Brafman et al. “Performance level profiles: A formal language for describing the expected performance of functional modules”. In: *IROS*. South Korea, 2016.
[9] Colin Archibald and Emil Petriu. “Skills-oriented robot programming”. In: *Int. Conf. on Intelligent Autonomous Systems*. Pittsburgh, PA, USA, 1993.
[10] Simon Bøgh et al. “Does your robot have skills?” In: *International Symposium on Robotics*. Taipei, Taiwan, 2012.
[11] Franz Steinmetz and Roman Weitschat. “Skill Parametrization Approaches and Skill Architecture for Human-Robot Interaction”. In: *CASE*. Fort Worth, TX, USA, 2016.

Dependable Skill-based executive layer

- Executive layer:
  - translation of decision *actions* into **executable** commands
  - abstraction of the functional layer
  - "task achieving behaviors"[2], "procedures"[3], "performance profiles"[4], "skills"[5,6,7]
  - **Skills** represent functions/services available in the functional layer

⇝ **Model-based** design of the skill-based executive layer

⇝ Tools/processes to support a **dependability** analysis



[6] Rodney Brooks. "A robust layered control system for a mobile robot". In: *IEEE Journal on Robotics and Automation* 2.1 (1986), pp. 14–23.
[7] Felix Ingrand et al. "PRS: A high Level Supervision and Control Language for Autonomous Mobile Robots". In: *ICRA*. Minneapolis, MN, USA, 1996.
[8] Ronen Brafman et al. "Performance level profiles: A formal language for describing the expected performance of functional modules". In: *IROS*. South Korea, 2016.
[9] Colin Archibald and Emil Petriu. "Skills-oriented robot programming". In: *Int. Conf. on Intelligent Autonomous Systems*. Pittsburgh, PA, USA, 1993.
[10] Simon Bøgh et al. "Does your robot have skills?" In: *International Symposium on Robotics*. Taipei, Taiwan, 2012.
[11] Franz Steinmetz and Roman Weitschat. "Skill Parametrization Approaches and Skill Architecture for Human-Robot Interaction". In: *CASE*. Fort Worth, TX, USA, 2016.

Skill Definitions

- Skills represent functions available in the functional layer
- Skills are modular and participate to the mission design
- Skills that share relations (dependencies, exclusion, . . .) are grouped into a Skillset
- These relations are modeled as Resources
- The correct execution of each skillset is managed by a Skillset Manager

[12] Charles Lesire et al. "Formalization of Robot Skills with Descriptive and Operational Models". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA, 2020

Skill Definitions

- Skills represent functions available in the functional layer
- Skills are modular and participate to the mission design
- Skills that share relations (dependencies, exclusion, . . .) are grouped into a Skillset
- These relations are modeled as Resources
- The correct execution of each skillset is managed by a Skillset Manager

Skillset formulation

Skillsets are described using a custom *Robot-language*[12].

[12] Charles Lesire et al. "Formalization of Robot Skills with Descriptive and Operational Models". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA, 2020

Skillset formulation

Skillset is built upon three types of elements:

- data — skillset shared data
- resource — shared resources within the skillset
- event — external means to interact with resources
- skill — robot skill definition

```
skillset name {
    data     { ... } // skillset data definition
    resource { ... } // skillset resources definition
    event    { ... } // skillset events definition
    skill    { ... } // skillset skills definition
}
```

Skill formulation

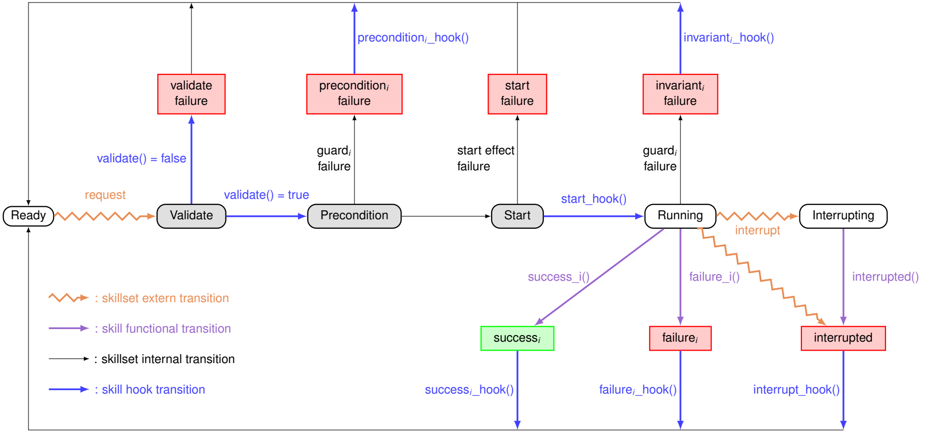
Robot Skill

Input	input data
Output	ouput data
Precond.	preconditions on resources
Start	effects on start
Invariant	resources invariants
Progress	progress of the running skill
Terminate	of the skill: effets & postcondition

Interrupt	interrupting the skill
Success	
Failure	

```
skill skill_name {
    input {...}
    output {...}
    precondition {...}
    start ...
    invariant {...}
    progress {...}
    interrupt ...
    success ...
    failure ...
}
```

Skill Execution Model state-machine



The skillset semantics

Skills Design

Goals

- **"Safe"** robots
  - clear/unambiguous specification
  - specification verification/analyses
  - real execution conforming to specification
- **Abstraction**
  - different robots can share skills
  - don't rewrite code
  - ease deployment and reconfiguration

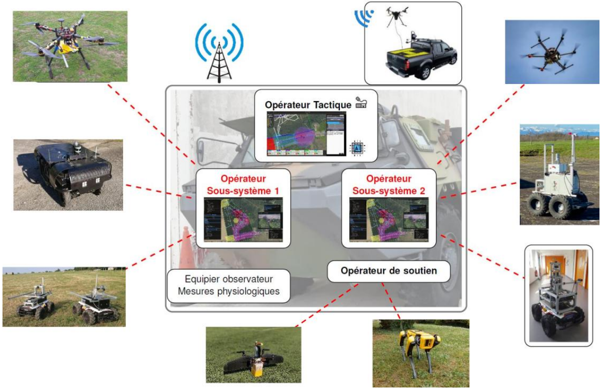
Principe

- "Formal" specification
  - "formal" language: the **Robot-language**
  - **standard interface** for the decision layer
  - various possible **verifications and analyses**
- ROS 2 code generator
  - don't introduce bugs
  - ensure that the real execution conforms to the model

Cohoma Challenge – Robotic platforms

10 platforms

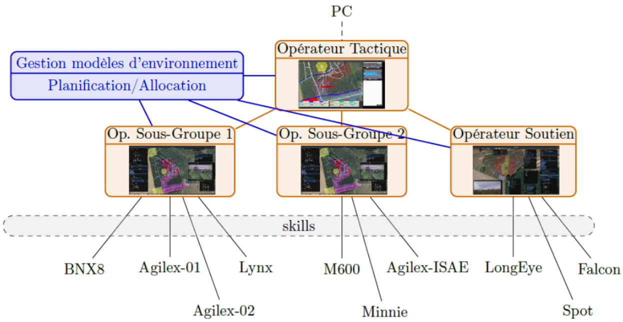
- 2 multi-rotor drones
- 1 captive drone
- 1 convertible drone
- 5 ground robots on wheels
- 1 quadruped robot



Cohoma Challenge – Robotic platforms

10 platforms

- 2 multi-rotor drones
- 1 captive drone
- 1 convertible drone
- 5 ground robots on wheels
- 1 quadruped robot



Code Generation

- Generation of a Skillset Manager Template
- . . . as a **ROS 2** node (and messages)
- All the logical behaviour is generated as:
  - resource management (guards, effects, events)
  - skills FSM (inputs/outputs, invariants, states)
- The developer still has to:
  - implement the `validate_hook` if they want to check something before starting
  - implement the **actual skill execution** in relation with the functional layer in between the `start_hook` and the `success` or `failure` terminations

Resolution Techniques

Assisting the design

Verifying the correctness of the skillset models using formal methods.

[13] Leonardo De Moura and Nikolaj Bjørner. "Z3: An efficient SMT solver". In: *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2008, pp. 337–340.

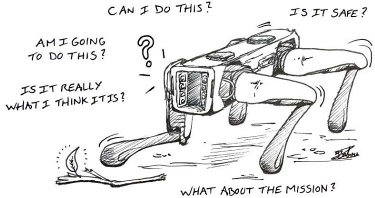
[14] Gabriela Catalán Medina et al. "A skill fault model for autonomous systems". In: *IEEE/ACM International Workshop on Robotics Software Engineering (RoSE)*. Pittsburgh, USA, 2022.

Resolution Techniques

Assisting the design

Verifying the correctness of the skillset models using formal methods.

- Formalism: encoding the formulæ in Satisfaction Modulo Theory (SMT)
  - Resources modeled as state machines
  - Guards (resource logical formulæ)
  - Effects (resource transitions)
- Solver used for inconsistencies check: Z3[13]



[13] Leonardo De Moura and Nikolaj Bjørner. "Z3: An efficient SMT solver". In: *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2008, pp. 337–340.

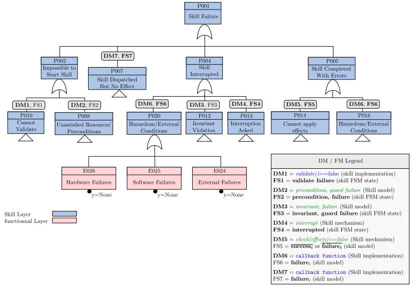
[14] Gabriela Catalán Medina et al. "A skill fault model for autonomous systems". In: *IEEE/ACM International Workshop on Robotics Software Engineering (RoSE)*. Pittsburgh, USA, 2022.

Resolution Techniques

Assisting the design

Verifying the correctness of the skillset models using formal methods.

- Formalism: encoding the formulæ in Satisfaction Modulo Theory (SMT)
  - Resources modeled as state machines
  - Guards (resource logical formulæ)
  - Effects (resource transitions)
- Solver used for inconsistencies check: Z3[13]



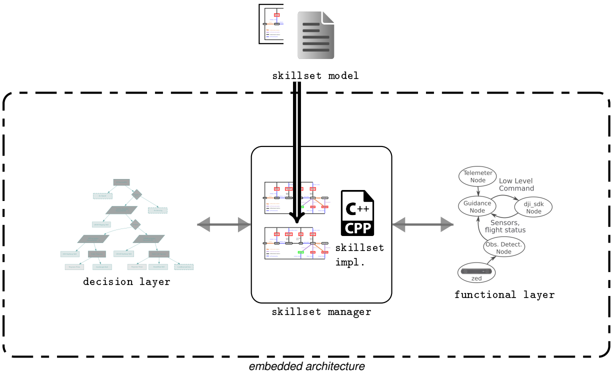
Fault Tree Analysis[14]

- Hazards might be combined using logic symbols
- Qualitative analysis can be performed on the skill fault tree.

[13] Leonardo De Moura and Nikolaj Bjørner. "Z3: An efficient SMT solver". In: *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2008, pp. 337–340.

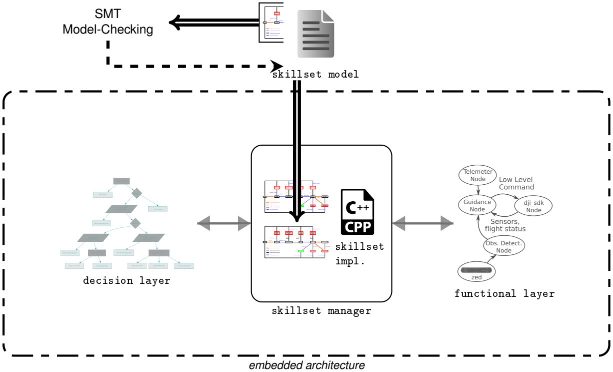
[14] Gabriela Catalán Medina et al. "A skill fault model for autonomous systems". In: *IEEE/ACM International Workshop on Robotics Software Engineering (RoSE)*. Pittsburgh, USA, 2022.

Dependable Skill-based executive layer



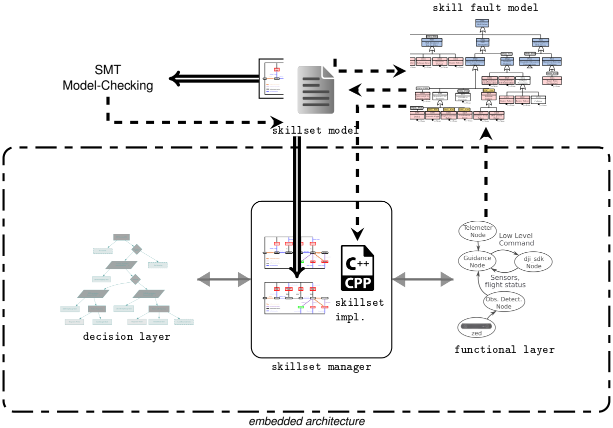
15 Alexandre Albore et al. "Skill-based design of dependable robotic architectures". In: *Robotics and Autonomous Systems* 160.104318 (2023)

Dependable Skill-based executive layer



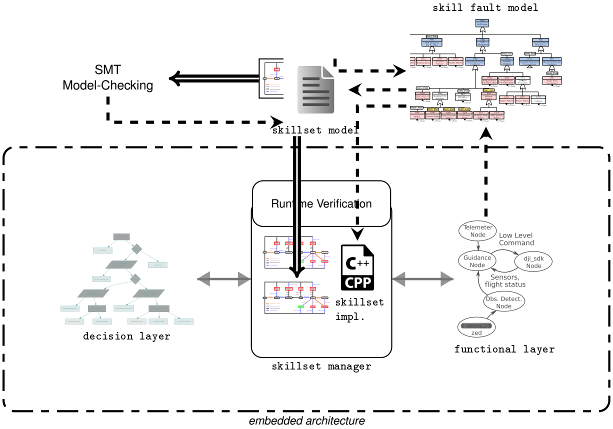
[15] Alexandre Albore et al. "Skill-based design of dependable robotic architectures". In: *Robotics and Autonomous Systems* 160.104318 (2023)

Dependable Skill-based executive layer



[15] Alexandre Albore et al. "Skill-based design of dependable robotic architectures". In: *Robotics and Autonomous Systems* 160.104318 (2023)

Dependable Skill-based executive layer



[15] Alexandre Albore et al. "Skill-based design of dependable robotic architectures". In: *Robotics and Autonomous Systems* 160.104318 (2023)

Decision layer

The decision-making layer is agnostic of the actual functional layer

Decision layer

The decision-making layer is agnostic of the actual functional layer

Thus it can be implemented in different ways, e.g.

- Behaviour trees
- Automated planning
- Actors managing goal cycles
- etc.

Decision layer

The decision-making layer is agnostic of the actual functional layer

Thus it can be implemented in different ways, e.g.

- Behaviour trees
- Automated planning
- Actors managing goal cycles
- etc.

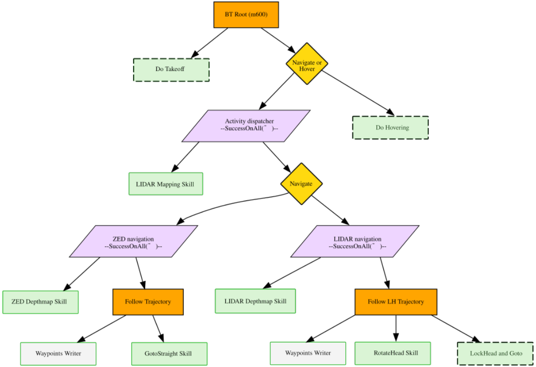
Some examples are coming. . .

Behaviour Trees

- Need to adapt in case of contingency
- E.g. for a UAV mission
  - React to a sensor failure using other on-board sensors and an alternative processing
  - Adapt the mission depending on exogenous events

Behaviour Trees

- Need to adapt in case of contingency
- E.g. for a UAV mission
  - React to a sensor failure using other on-board sensors and an alternative processing
  - Adapt the mission depending on exogenous events
- Behaviour trees allow to specify nominal and degraded plans



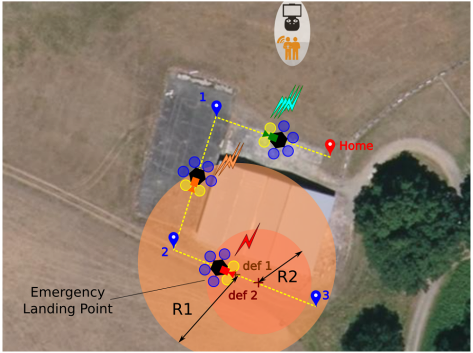
UAVs Mission Reconfiguration

- 3D Mapping of a storage building/infrastructure inspection
- BVLOS scenario (Beyond Visual Line of Sight of the teleoperator)[16]

[16] Alexandre Albore et al. "Skill-Based Architecture Development for Online Mission Reconfiguration and Failure Management". In: *IEEE/ACM International Workshop on Robotics Software Engineering (RoSE)*. Madrid, Spain, 2021

UAVs Mission Reconfiguration

- 3D Mapping of a storage building/infrastructure inspection
- BVLOS scenario (Beyond Visual Line of Sight of the teleoperator)[16]
- **Contingency**: Communication Loss
- **Reconfiguration**:
  - React to a sensor failure using other on-board sensors and an alternative processing
  - Adapt the mission depending on exogenous events
- more concretely:
  1. adapt bandwidth
  2. land if communication lost



[16] Alexandre Albore et al. "Skill-Based Architecture Development for Online Mission Reconfiguration and Failure Management". In: *IEEE/ACM International Workshop on Robotics Software Engineering (RoSE)*. Madrid, Spain, 2021

Earth Observation Satellites (EOS)

- An EOS is a spacecraft designed for gathering information about the Earth from orbit
- They point to targets and then take images with a sensor (visible, IR, radio, etc.)
- The orbit determines which areas are visible at a given time

- Observation tasks depend on the environment not known in advance, e.g. cloud cover, etc.
- Platform constraints (Attitude control, energy consumption...)
- New customer's requests

Earth Observation Satellites (EOS)

- An EOS is a spacecraft designed for gathering information about the Earth from orbit
- They point to targets and then take images with a sensor (visible, IR, radio, etc.)
- The orbit determines which areas are visible at a given time

- Observation tasks depend on the environment not known in advance, e.g. cloud cover, etc.
- Platform constraints (Attitude control, energy consumption...)
- New customer's requests

Need to embed some functions on-board
in order to exploit data only available on-board

A decision layer based on Actors

The decision layer is built upon the OARA framework[17](https://onera-robot-skills.gitlab.io):

- **Architectural framework** to design and manage the **deliberative reasoning** of an autonomous system.
- Based on a **hierarchy of actors** managing planning and acting through **goal decomposition**.

[18] Charles Lesire et al. "A Hierarchical Deliberative Architecture Framework based on Goal Decomposition". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022)*. Kyoto, Japan, 2022.

A decision layer based on Actors

The decision layer is built upon the OARA framework[17](https://onera-robot-skills.gitlab.io):

- **Architectural framework** to design and manage the **deliberative reasoning** of an autonomous system.
- Based on a **hierarchy of actors** managing planning and acting through **goal decomposition**.

Derived from:

- Actors [18]: Reasoning components that manage planning and acting of a specific task.
- The concept of goal lifecycle[19]
- The Hierarchical Task Network (HTN) model and the Partial-Order Planning approach.

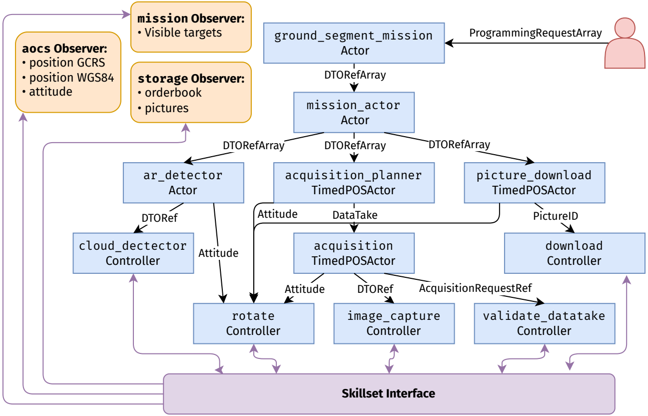
---

[18] Charles Lesire et al. "A Hierarchical Deliberative Architecture Framework based on Goal Decomposition". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022)*. Kyoto, Japan, 2022.

[19] Malik Ghallab et al. "The actor's view of automated planning and acting: A position paper". In: *Artificial Intelligence* 288 (2014), pp. 1–17.

[20] Mark Roberts et al. "Goal Reasoning, Planning, and Acting with ACTOR SIM, The Actor Simulator". In: *Annual Conference on Advances in Cognitive Systems*. Evanston, IL, USA, 2016.

Decision layer architecture for EOS

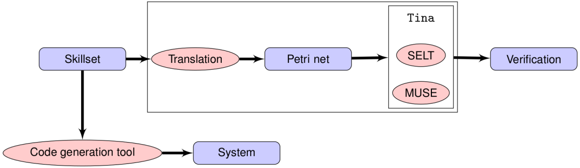


[21] Alexandre Albore and Rafael Bailon-Ruiz. "A Hierarchical Temporal Mission Controller for an Autonomous Earth Observation Satellite". In: *Multi-Agent Systems for Space Applications Workshop (AAMAS)*. Auckland, NZ, 2024.

Conclusions

Skills composition and verification[22]

- Skills can be composed
- and translated to Petri Nets
- and verified with adapted tools (Tina suite[23])



[23] Baptiste Pelletier et al. "SkiNet, A Petri Net Generation Tool for the Verification of Skillset-based Autonomous Systems". In: *International Workshop on Formal Methods for Autonomous Systems.* Berlin, Germany, 2022.

[24] Bernard Berthomieu et al. "The tool TINA – construction of abstract state spaces for Petri nets and time Petri nets". In: *International journal of production research* 42.14 (2004), pp. 2741–2756.

Conclusions

Warehouse robotic management

- Scaling up challenge ($\sim$ ~~800~~ *10k* autonomous UGVs)
- Increasing resilience
- Moving from a centralized architecture to a distributed one
- Advancing on the MAPF front

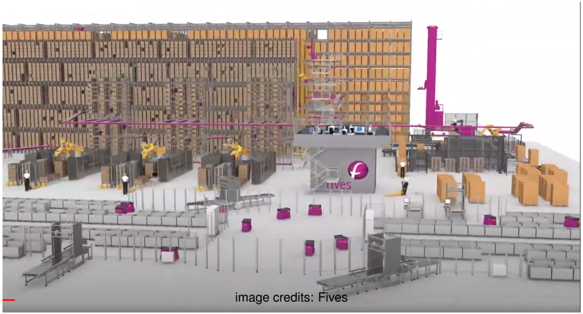
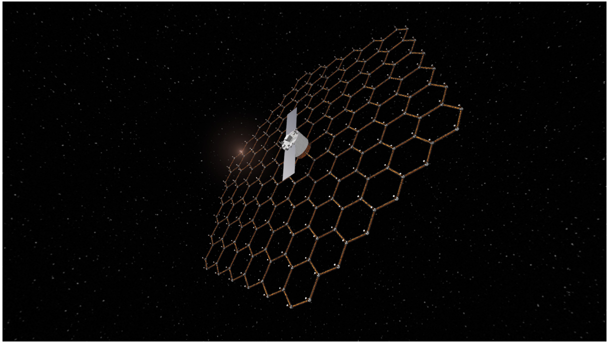


image credits: Fives

Conclusions

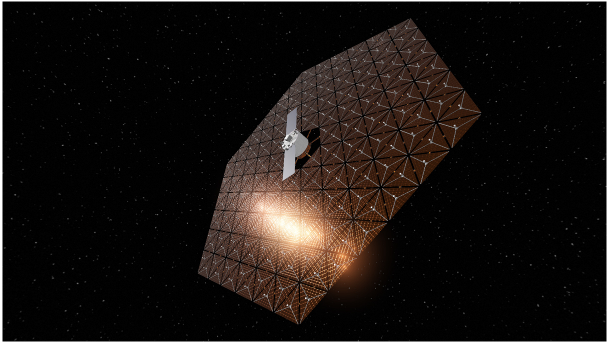


On-orbit satellite assembly

- **Crawling robots** moving on a truss beam lattice structure
- **Automated planning**-based approach[25]
- Advancing on the MAPF front

[25] Alexandre Albore and Mathieu Rognant. "Integrated Modeling and Planning for On-Orbit Assembly of Large Space Structures with Mobile Crawling Robots". In: *Multi-Agent Systems for Space Applications Workshop (AAMAS)*. Auckland, NZ, 2024.

Conclusions



On-orbit satellite assembly

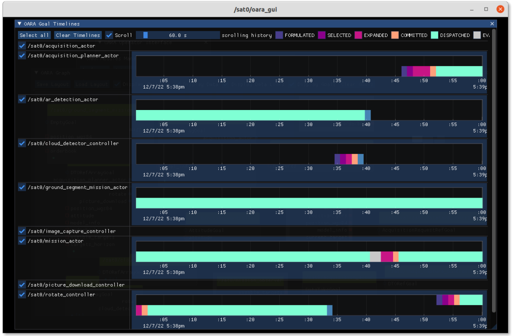
- **Crawling robots** moving on a truss beam lattice structure
- **Automated planning**-based approach[25]
- Advancing on the MAPF front

[25] Alexandre Albore and Mathieu Rognant. “Integrated Modeling and Planning for On-Orbit Assembly of Large Space Structures with Mobile Crawling Robots”. In: *Multi-Agent Systems for Space Applications Workshop (AAMAS)*. Auckland, NZ, 2024.

Conclusions

Mission Planning for EOS

- Need to **produce plans for a defined horizon**, steering the satellite towards observation targets
  - Time component of actions is critical
  - Expressiveness of Hierarchical Task Network Planning → **HDDL 2.1** [26]



[26] Damien Pellier et al. “HDDL 2.1: Towards Defining a Formalism and a Semantics for Temporal HTN Planning”. In: *Proceedings of the International Workshop of Hierarchical Planning (ICAPS)*. Prague, 2023.

Thank you!