Final Workshop

Towards Smart Autonomous Cyber-Physical Systems: Unmanned Aerial/Ground Vehicles and Robots

Modelling & Code Generation

Etienne Brosse, Softeam
Micha Sende, Lakeside Labs
Gianluca Prato, LINKS Foundation

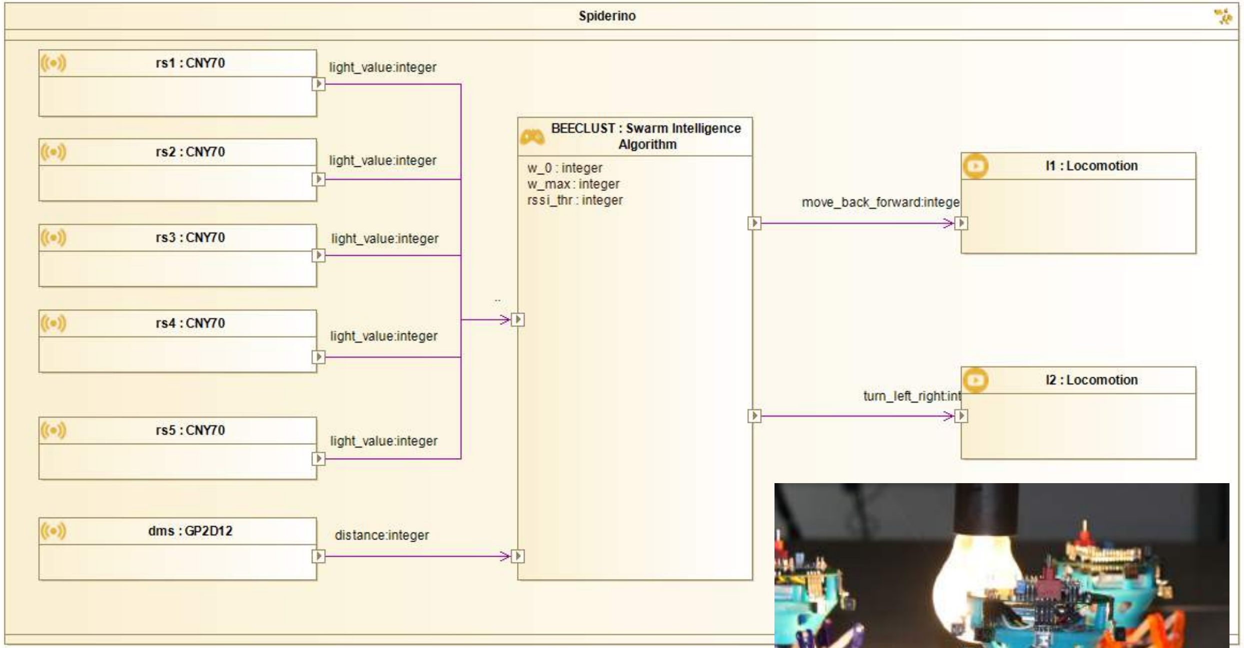
Turin, December 13th 2019

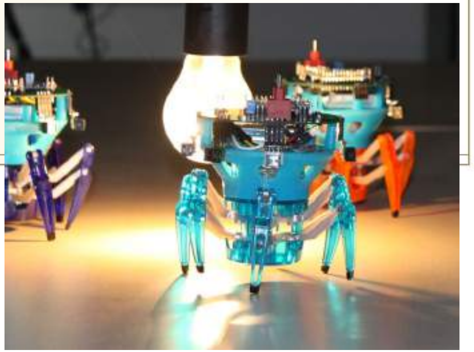
CPS Models

Available CPS Models

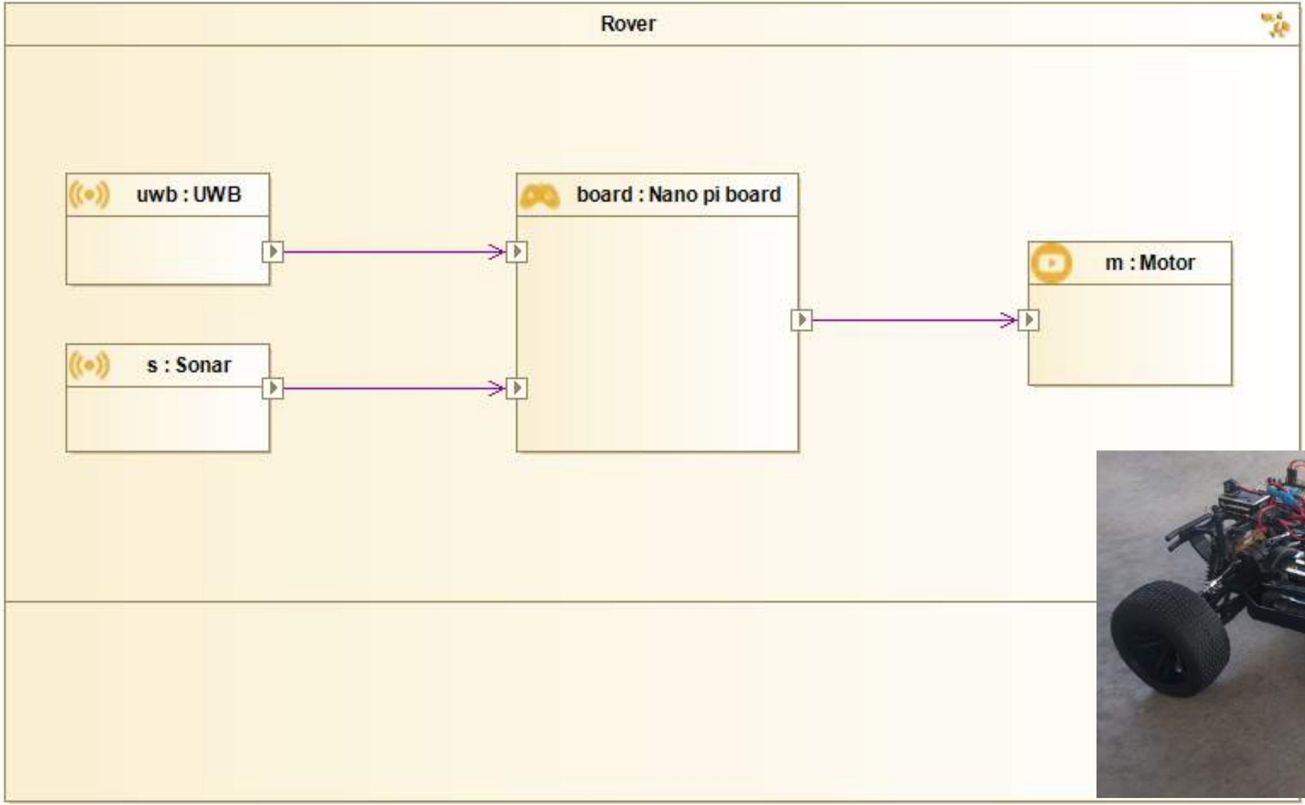
CPS and Human Interaction Modeling

- **Spiderino**
  - **Hardware only**
- **Drone**
  - **Hardware and software**
- **Rover**
  - **Hardware and software**
- **Logistic**
  - **Hardware and software.**

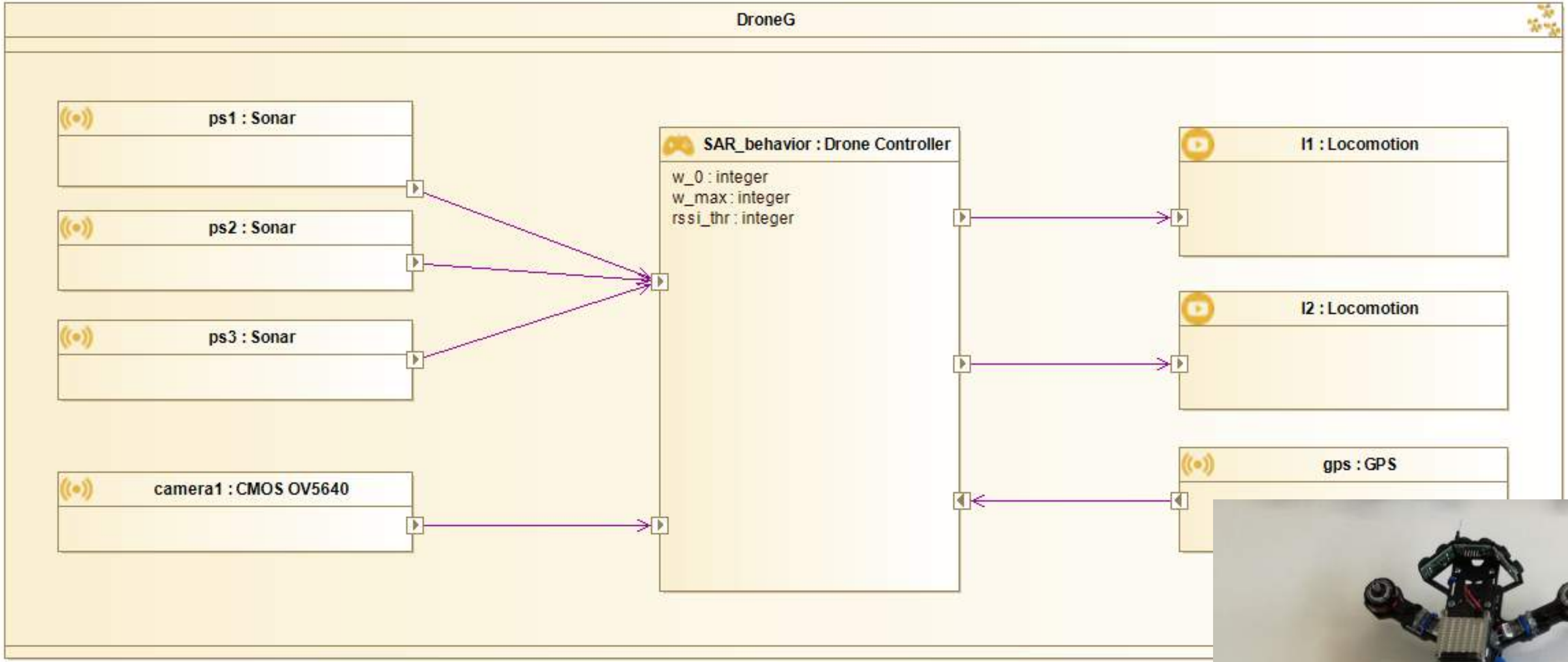


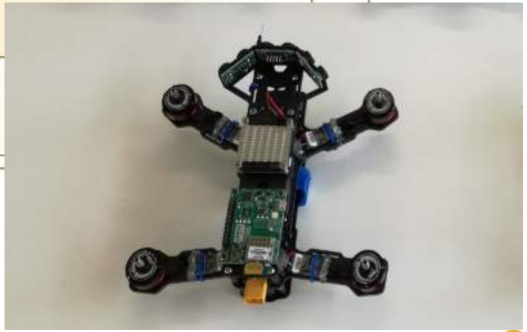


Rover– SAR Demo



Drone– SAR demo

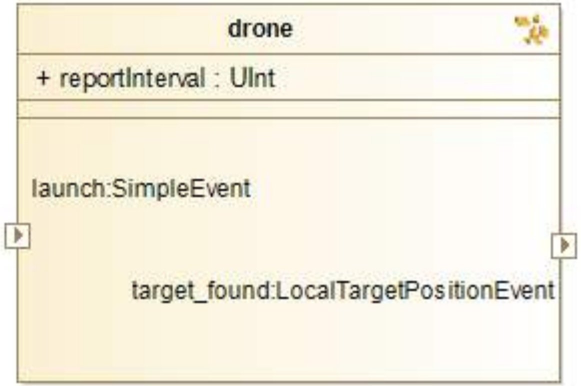




Communication

CPS and Human Interaction Modeling

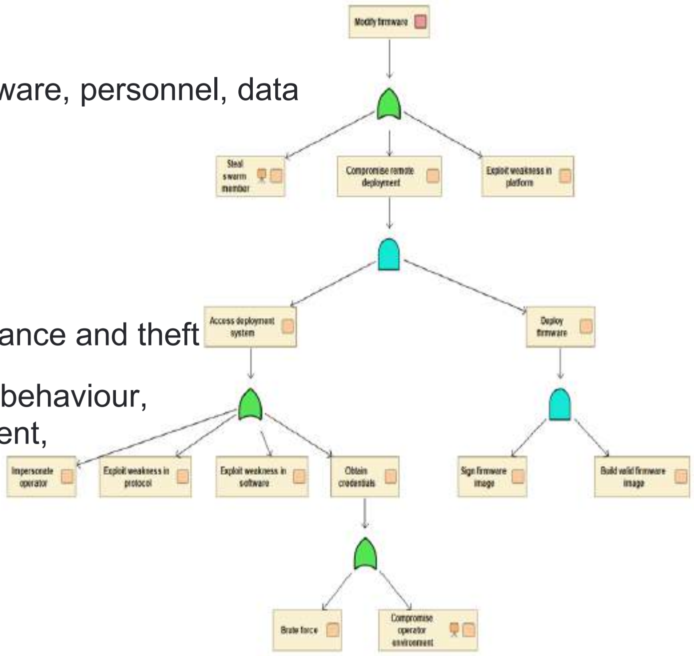
Communication can be modeled for each swarm member and each parameter




```
# Configure endpoint
#
endpoint = {
  name = "drone"        # Possibly non-unique name for the local node
  deviceClass = "drone"   # Discoverable device class
  type = "zyre"          # Endpoint type
  parameters = {         # Endpoint parameters, which for Zyre endpoints can be:
    # ifname = "eth0"      # Network interface to bind to
    # port = 34000         # Port to use for UDP beacons
  }
}
```

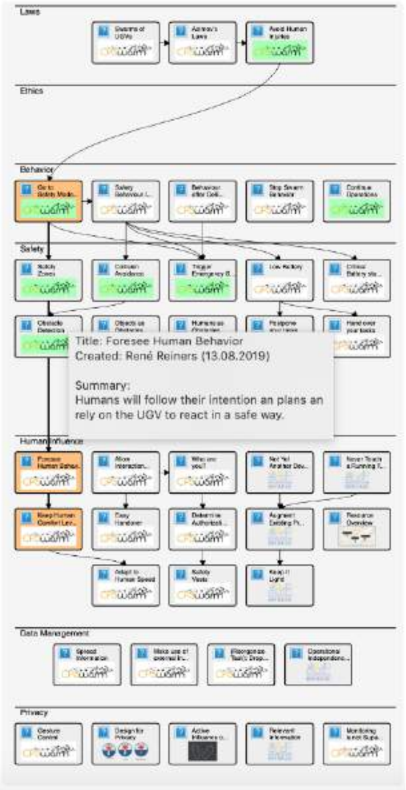
Security Modeling

- Assets
  - Generic assets: information, service, environment / software, hardware, personnel, data
  - Analysis of relevant assets
  - Use-case specific assets
- Attack trees
  - Attacker goals: cause sabotage, damage, physical harm or disturbance and theft
  - Methods: damage, destroy or redirect member, take advantage of behaviour, modify mission, compromise communication or operator environment, impersonate swarm member or operator, modify firmware
  - Use case specific attack trees
- Countermeasures
  - General countermeasures
  - Use case attacks and their mitigations
- Risk assessment: risk matrix

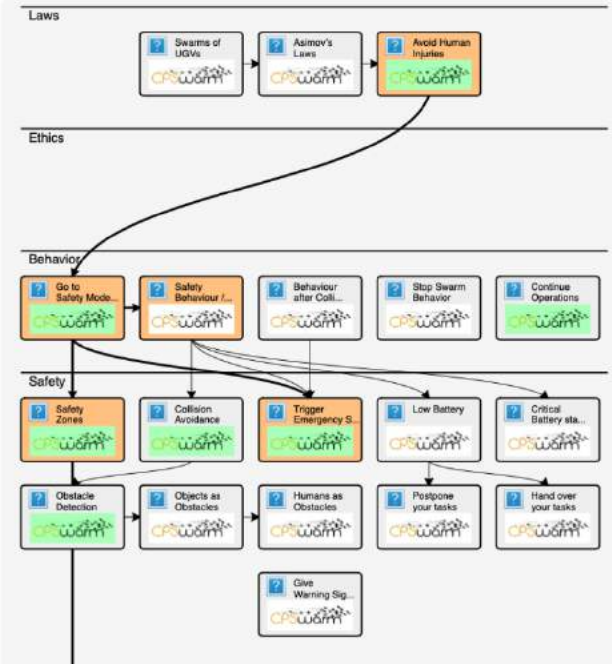


	Severity		
Likelihood	1	2	3
3: Certain	Medium	High	High
2: Likely	Low	Medium	High
1: Unlikely	Low	Low	Medium

Knowledge Input for Implementing Safety via Design Patterns



CPSwarm
Pattern Library



Connected patterns on Laws,
Behavior and Safety

Detailed Pattern Description

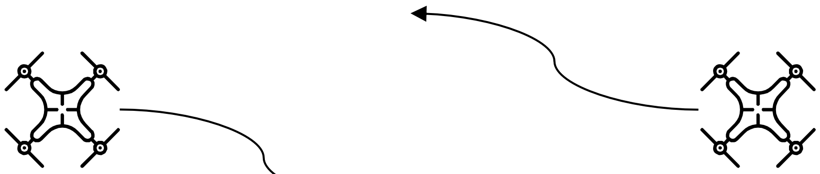
Swarm Modeling

Swarm Modeling

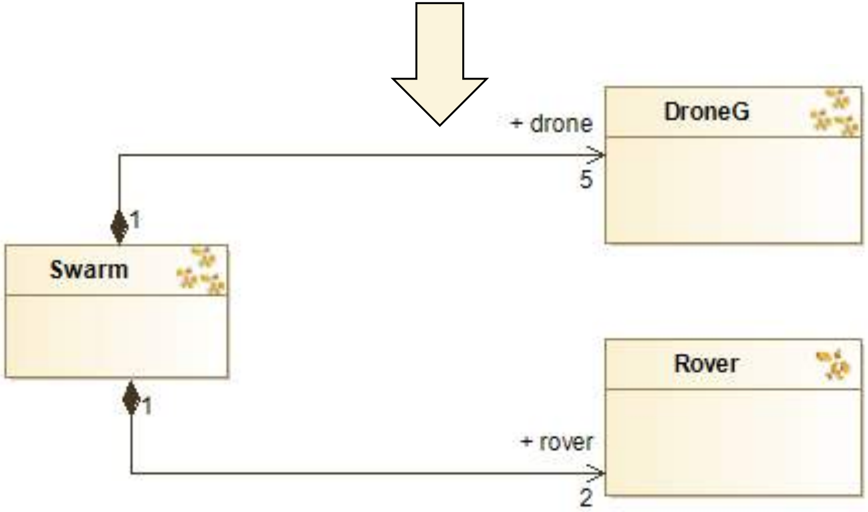
Swarm composition

- Number of swarm members
- Type of swarm members

Behavior of swarm members



Swarm Composition



Swarm Intelligence

AGENT

Simple behavior

Local information

Local interactions

Emergence →

SWARM

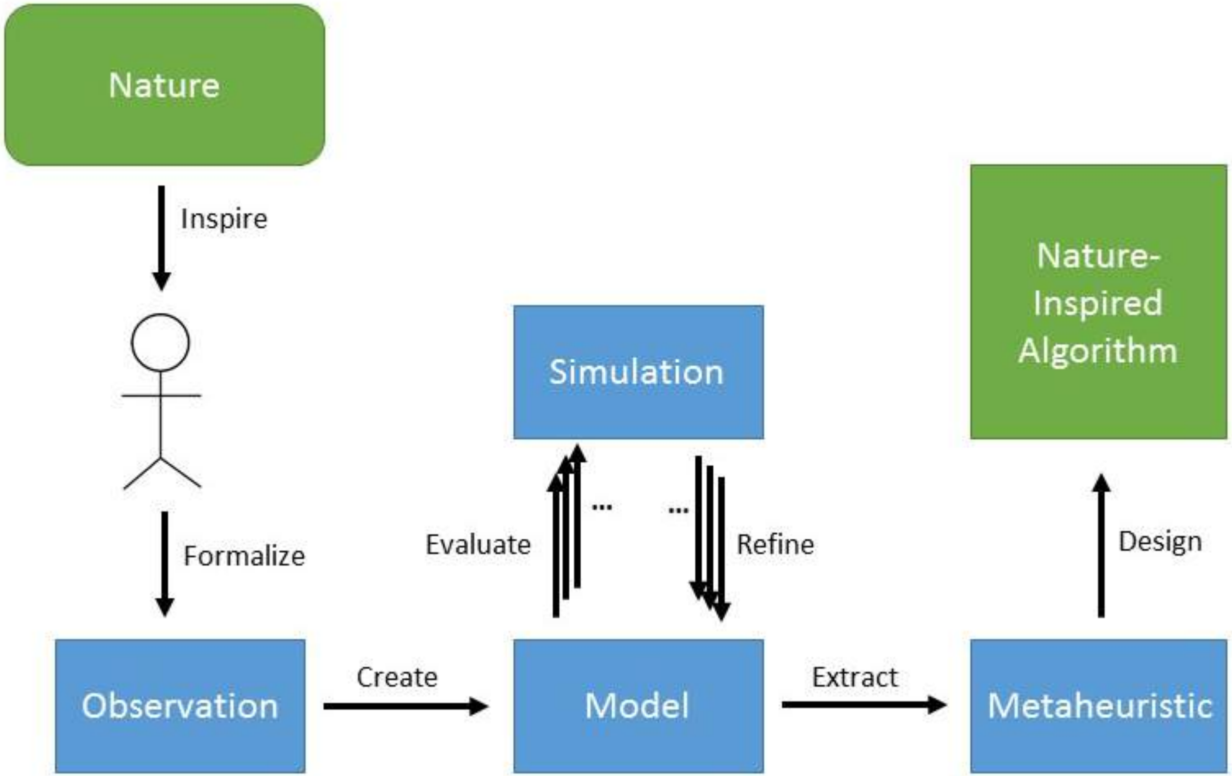
Many individuals

Homogeneous

Decentralized

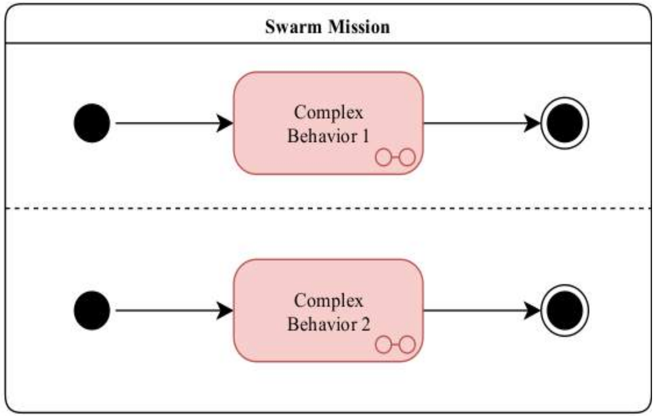
Self-organized

Biological Inspiration

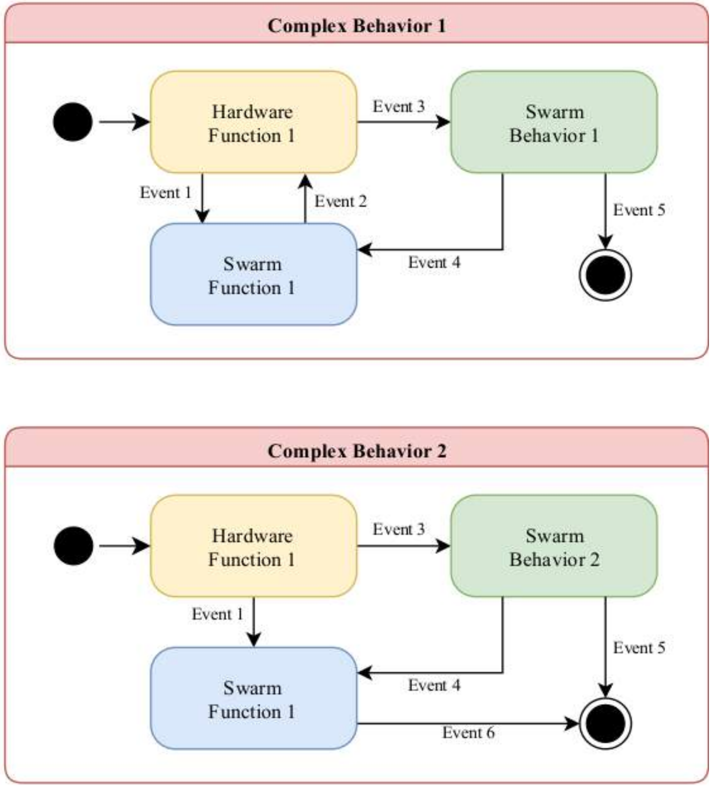


Behavior Model Hierarchy

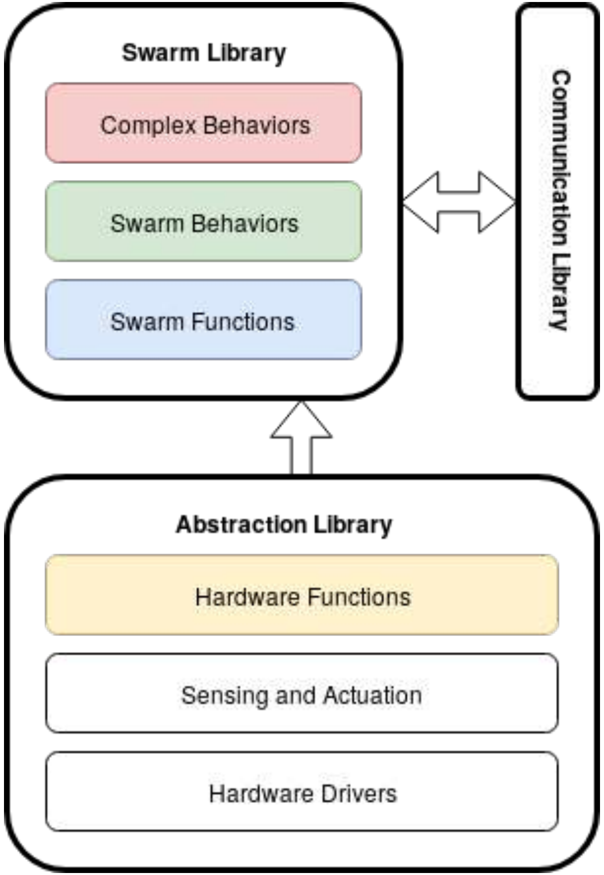
L_1

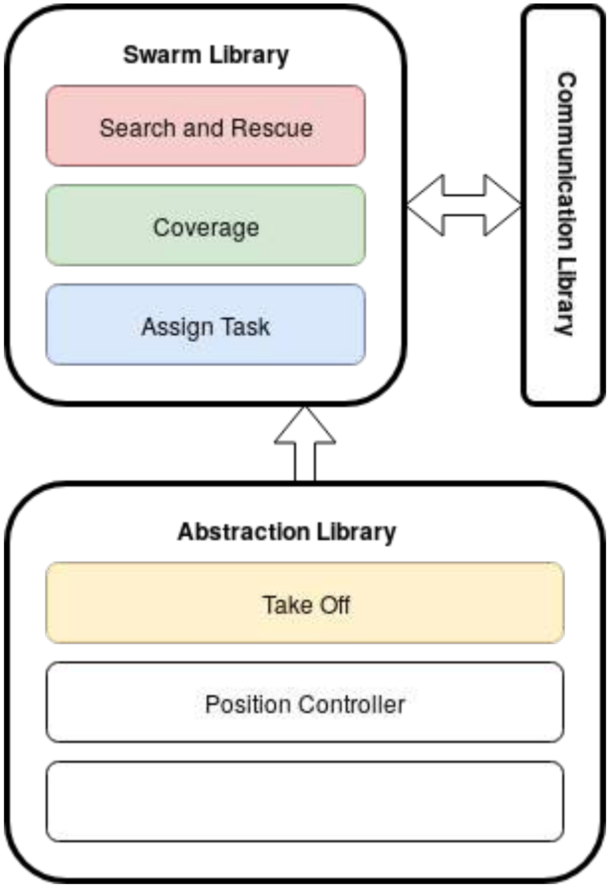


L_2

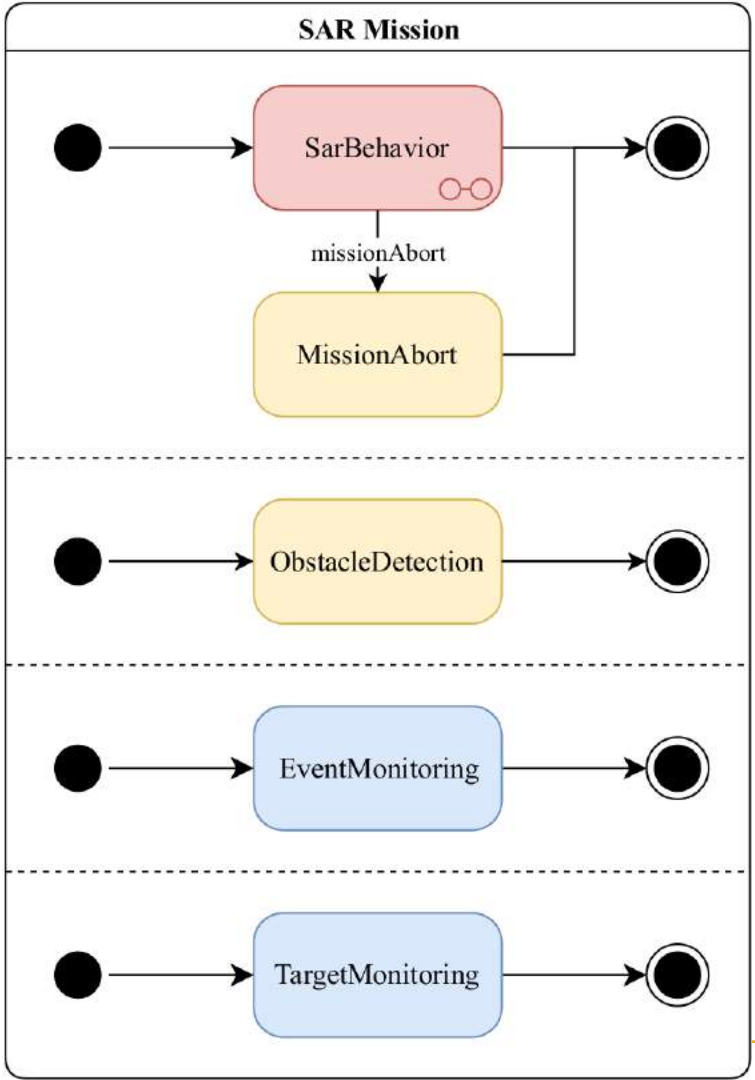


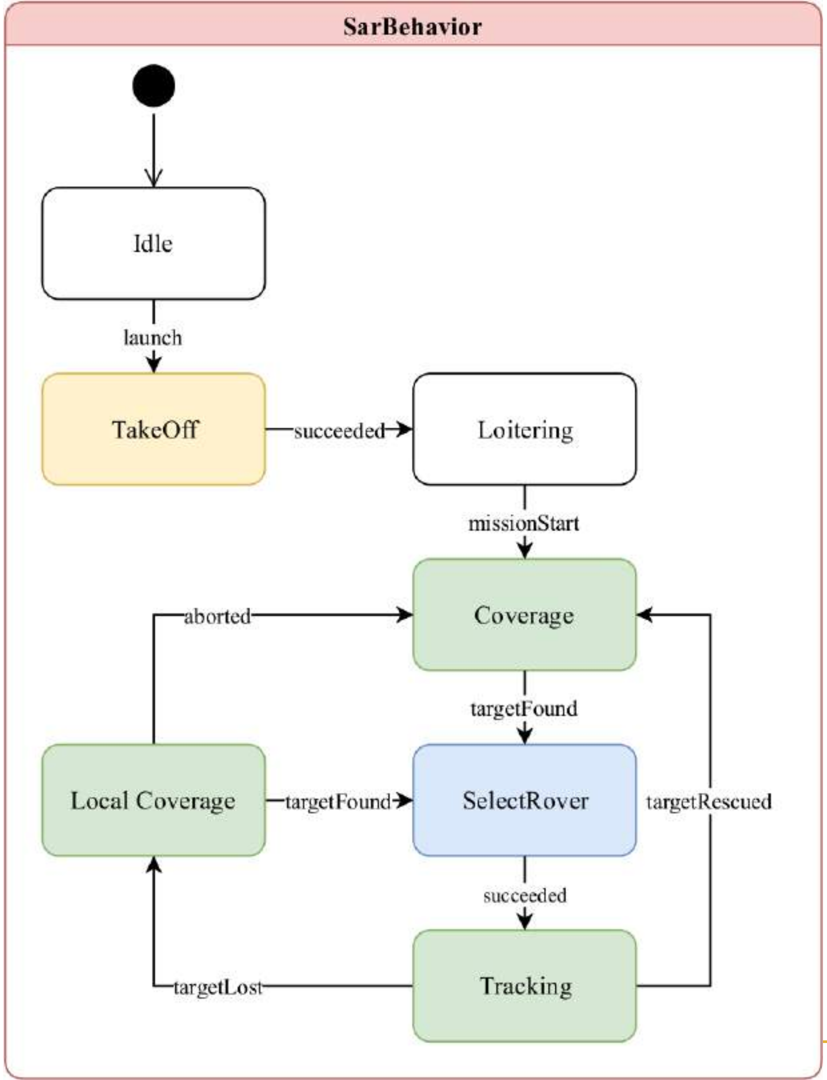
Behavior Library






www.cpswarm.eu
2 DRONES
3 TARGETS
2 ROVERS


SAR Mission
SarBehavior
missionAbort
MissionAbort
ObstacleDetection
EventMonitoring
TargetMonitoring


SarBehavior
Idle
launch
TakeOff
succeeded
Loitering
missionStart
Coverage
aborted
targetFound
Local Coverage
targetFound
SelectRover
targetRescued
succeeded
Tracking
targetLost

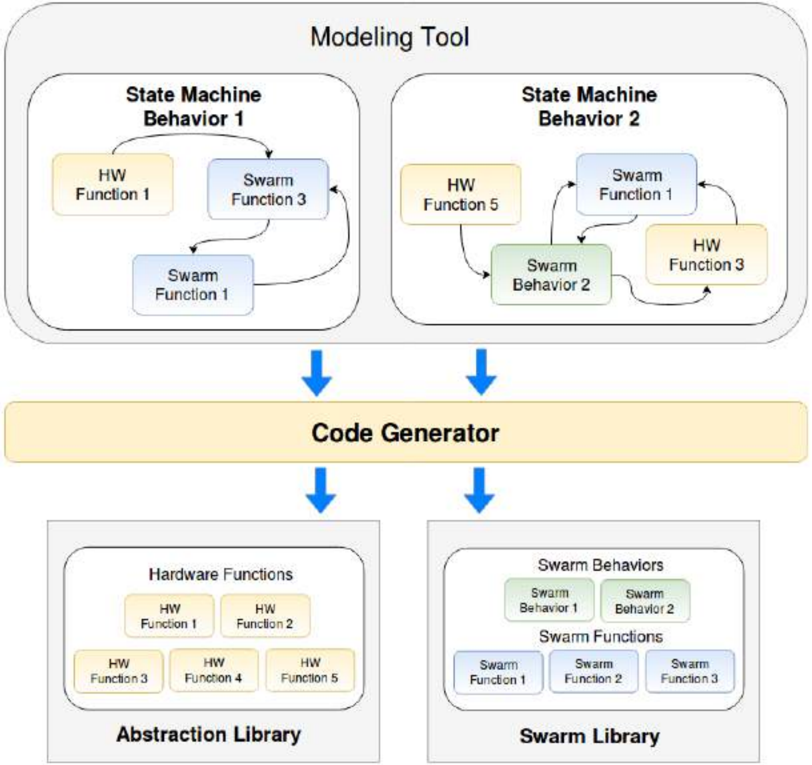
Automatic Code Generation

Automatic Code Generation

- The promise of modelling is to **shift** the focus from implementation to design
- Models can also be input for **code generators**.
- Automated code generation is a challenging task in software engineering but brings with it some **benefits**:
  1. Productivity
  2. Complexity hiding
  3. Portability
  4. Consistency and error rate reduction
- **Full code generation** is possible when fitted to the requirements of a specific domain

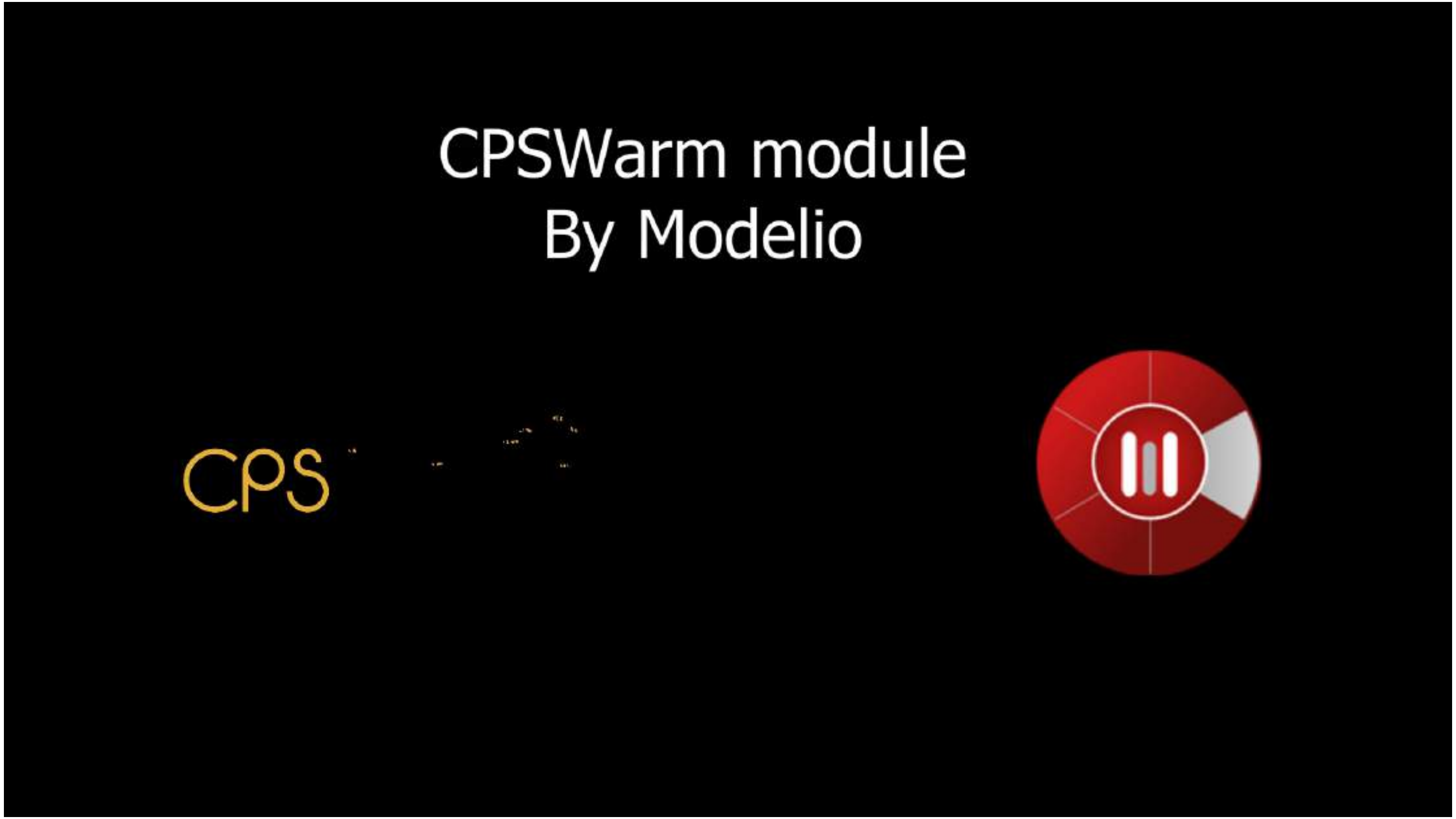
CPSwarm Code Generation

- CPSwarm Code Generator aims **not to substitute** developer work but to give support during the development
- Translate the model of a **Finite State Machine** (FSM) to actual deployable code
- Accepts the model of a FSM as **SCXML** file and produces ROS compatible code
- The generation process is realized using a **template-based** technique
- **Velocity** template engine is used to fill templates with data extracted from the SCXML input.



Modeling Video

Modeling example



THANKS!
ANY QUESTIONs?

CPSwarm

Final Workshop

Towards Smart Autonomous Cyber-Physical Systems:
Unmanned Aerial/Ground Vehicles and Robots

Etienne Brosse (SOFTEAM)

Micha Sende (Lakeside Labs)

Gianluca Prato (LINKS Foundation)

Contacts:

etienne.brosse@softeam.fr, sende@lakeside-labs.com,

gianluca.prato@linksfoundation.com

Website: https://www.cpswarm.eu

The projects leading to this application have received funding from the European Union's Horizon 2020 research and innovation program

CPSwarm

https://www.cpswarm.eu

Follow @CPSwarm_EU

Coordinator

LINKS

Partners

Fraunhofer FIT · Lakeside Labs · TTTech · Alpen-Adria Universität · Robotnik · SOFTEAM Cadextan · DigiSky · SEARCH-LAB

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731946.

CPSwarm Code Generation

- There are many possible ways to design a code generation pipeline. Basically we need to define two elements:
  - **Input**: where the information to be used in code generation comes from
  - **Output**: the final result of the generation task.