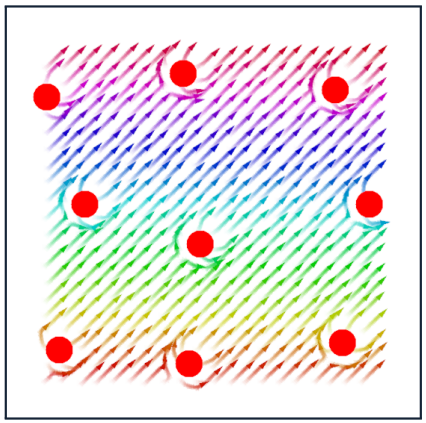
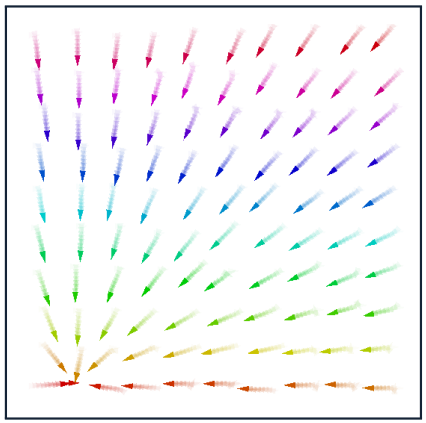
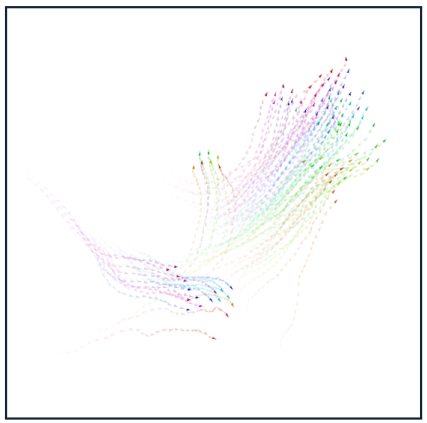
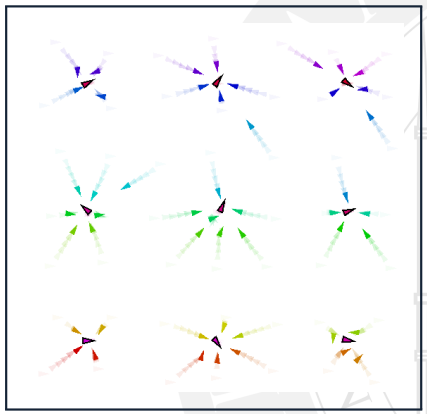
MacroSwarm
Field-based Compositional Framework for Swarm Programming

Gianluca Aguzzi gianluca.aguzzi@unibo.it
Roberto Casadei roby.casadei@unibo.it
Mirko Viroli mirko.viroli@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum – Università di Bologna

Talk @ COORDINATION 2023

Context

Networked Mobile Nodes + **Collective Behaviors**

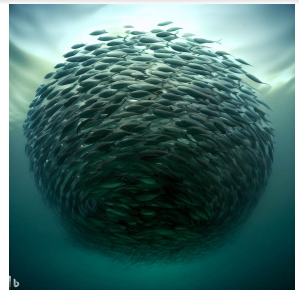
Swarm Behaviors

A **swarm behavior** is a *collective* behavior that emerges from the *local* interactions of a *population* of *autonomous* entities.

Inspiration from Nature: Social Animals

- *Ants* ➔ *foraging* and *nest building*
- *Fishes* ➔ *schooling*
- *Bees* ➔ *swarming*
- . . . But now let us focus on *engineering*! :)





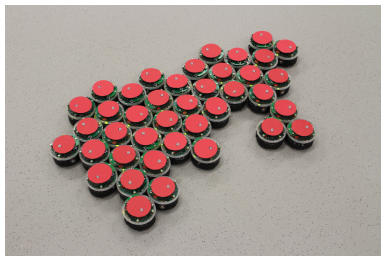
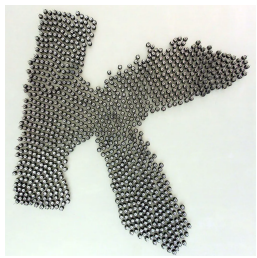
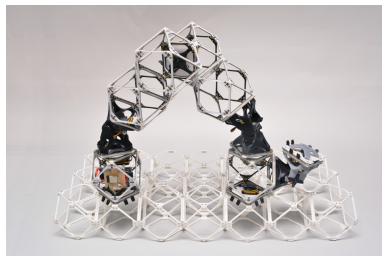
In a nutshell

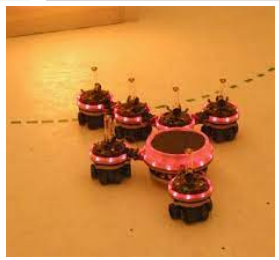
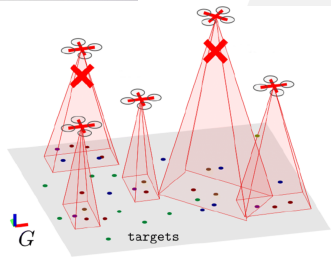
- **Emergent** ➔ *self-organisation* and *self-adaptation*
- **Decentralized** ➔ *locality* and *scalability*
- **Asynchronous** ➔ *robustness* and *fault-tolerance*

Swarm Behaviors: Taxonomy [1]

Main classes

- **Spatial Organization**: collective movements that lead to global spatial patterns
  - Collective structures ➔ functional to achieve certain goals
  - **Aggregation**, **pattern formation**, **self-assembly**, **object clustering**
- **Navigation**: coordinated motion in order to reach a target/perform collective tasks
  - **Exploration**, **motion**, **transportation**, **localization**
- **Decision-making**: lead the system to reach a global decision
  - **consensus**, **task allocation**, **group size regulation**, **collective perception**, ...

[1] Melanie Schranz et al. "Swarm robotic behaviors and current applications". In: *Frontiers in Robotics and AI* 7 (2020), p. 36

Programming Swarm Behaviors – Related Works

Problems

- How to express *swarm behaviors*[a]? ➔ *programming abstractions*
  - Move the viewpoint from *individual* to *collective* level ➔ macro-programming
- *Complexity* ➔ *collective* and *individual* levels ➔ **programming the emergence**[b]
- *Scalability* ➔ avoiding *centralized* and *global* approaches

[a] Gregory Mone. "Rise of the swarm". In: *Communications of the ACM* 56.3 (Mar. 2013), pp. 16–17

[b] Franck Varenne et al. "Programming the emergence in morphogenetically architected complex systems". In: *Acta biotheoretica* 63 (2015), pp. 295–308

Proposed solutions

- Orchestration-based: centralized approaches for expressing swarm behaviours
  - TeCoLa, Dolphin, PARoS
- Choreograph-based: decentralized approaches for expressing swarm behaviours
  - Buzz, Voltron, Meld

Limitations

- Lack of modularity, no formal semantics, no practical implementation, ...

MacroSwarm

MacroSwarm[2] is based on a macro programming approach called *Aggregate Computing*[3] (aka *field-based coordination*) for programming swarm behaviors in a modular and scalable way.

Features

- Cover the *main classes* of swarm behaviors
- *modular* and *composable* behaviors
- purely functional transformations from input to actuation fields ➔ easy to reason about
- Support several organization patterns ➔ flock-based, leader-based, ...
- Robust and scalable ➔ *decentralized* and *asynchronous*

[2] https://github.com/AggregateComputing/experiment-2023-coordination-swarm-behaviour

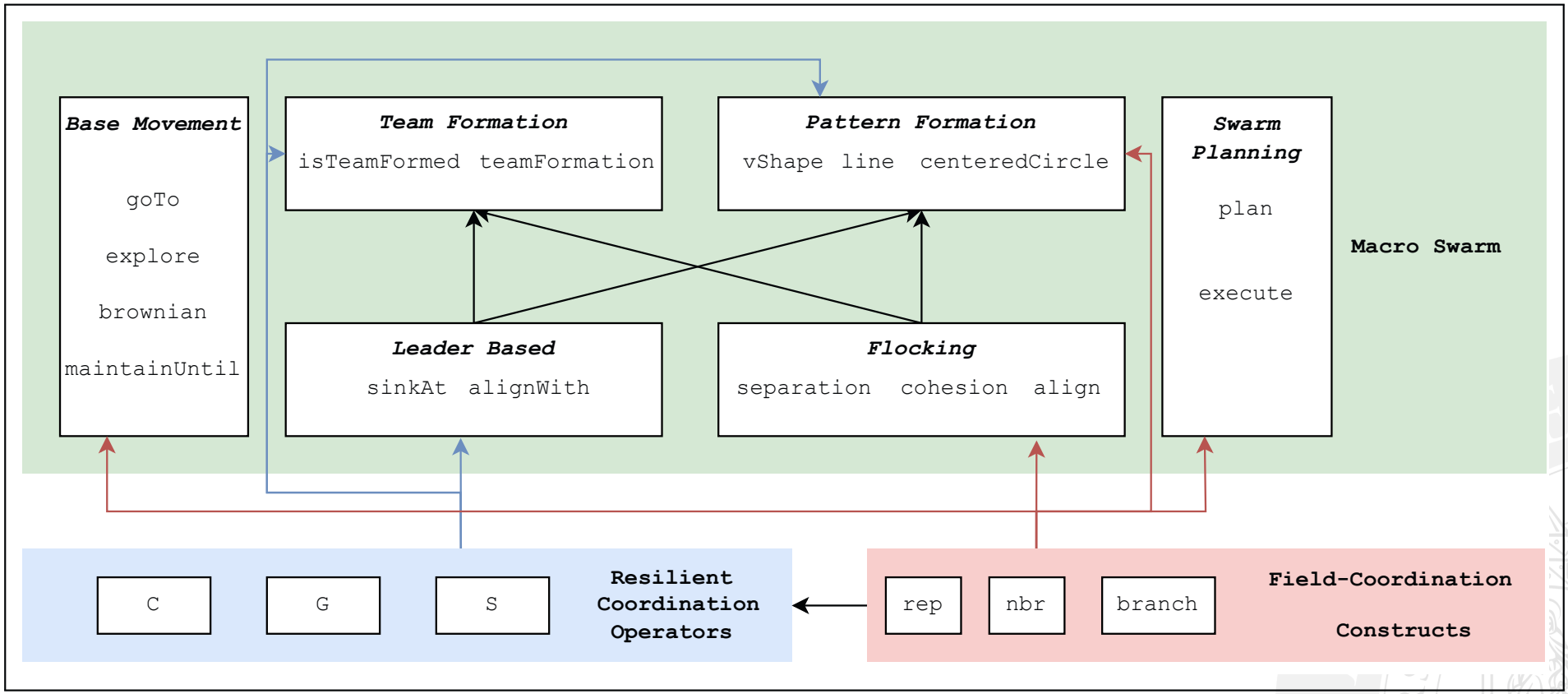
[3] Mirko Viroli et al. "From field-based coordination to aggregate computing". In: *Coordination Models and Languages: 20th*. Springer. 2018, pp. 252–279

Aggregate Computing I

Why

- *Top-down behaviour-based design* ➔ *compositionality* and *collective stance* of aggregate computing;
- *Scalability* ➔ fully decentralized and asynchronous
- *Formal approach* ➔ based on the field calculus
- *Pragmatism* ➔ witnessed by open-source, maintained, concrete software artifacts like the ScaFi DSL, Alchemist and ScaFi-Web
- *Operational flexibility* ➔ supporting different architectural styles and execution policies.

Architecture

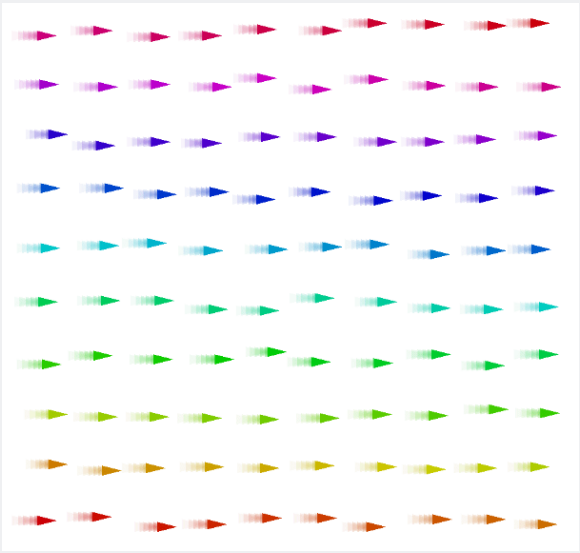


API Overview: Basic Movements

Control the movement of the individual agents within the swarm

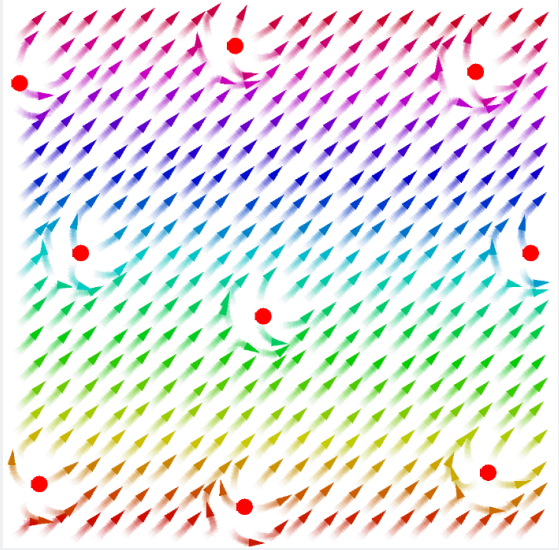
Constant movement

```
Velocity2D(0, 1)
```



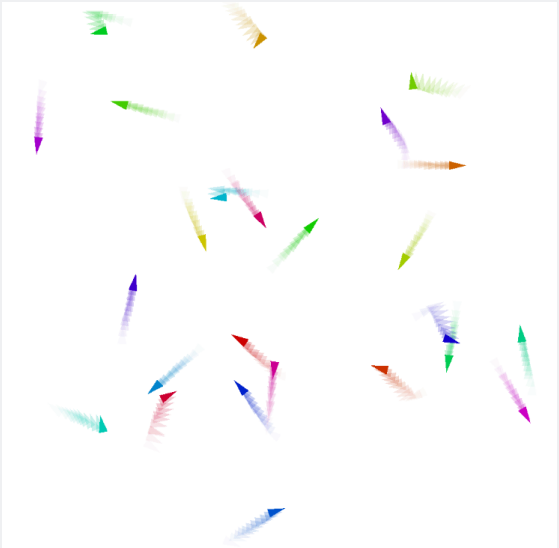
Obstacle avoidance

```
obstacleAvoidance(...)
```



Explore

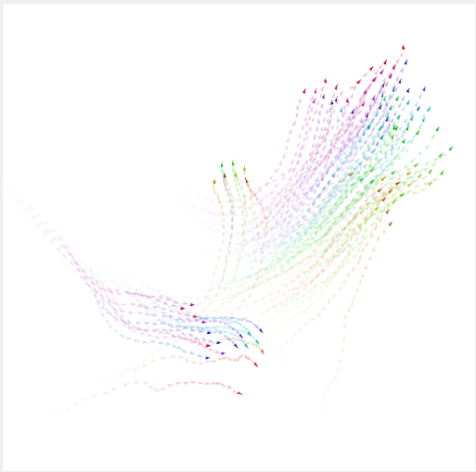
```
explore(...)
```



API Overview

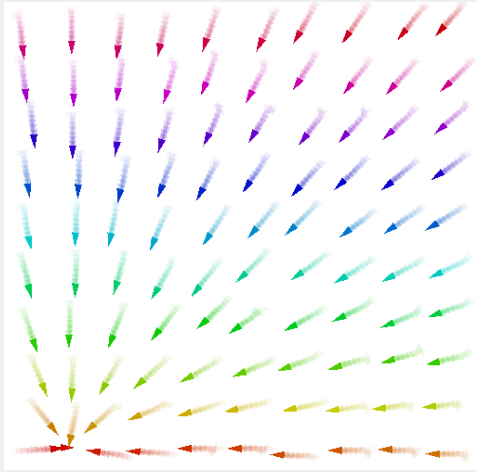
Flocking *Reynolds 1987*

```
rep(Zero)(v => reynold(v, ...))
```



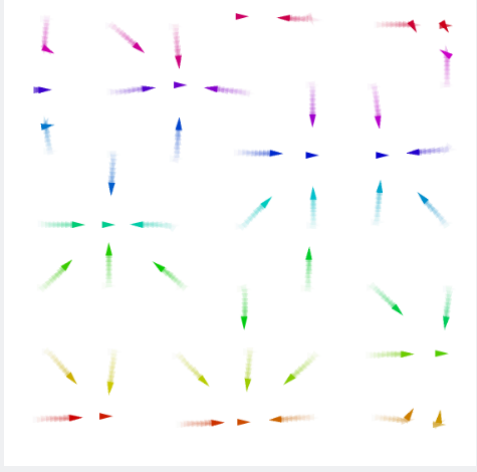
Leader Based *Gu et al. 2009*

```
sinkAt(leader) // boolean field
```



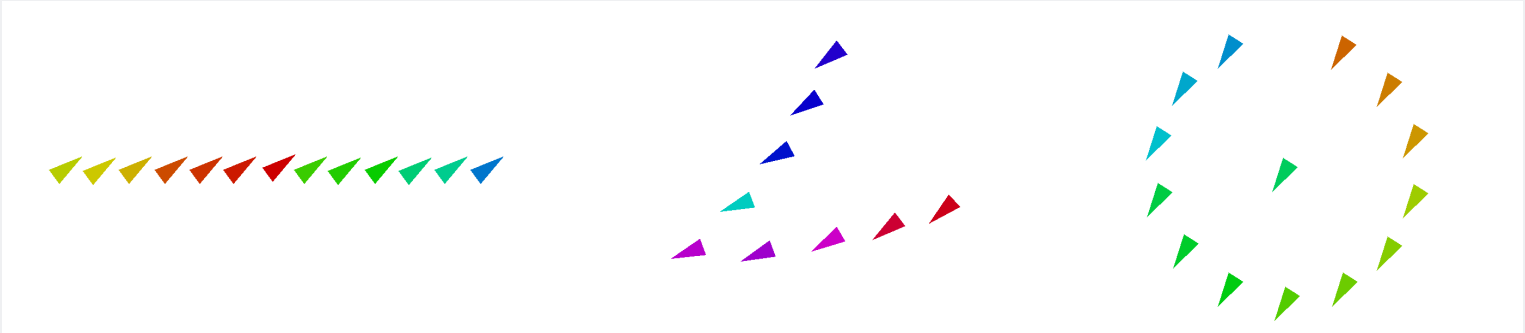
Team Formation

```
teamFormation(...)
```



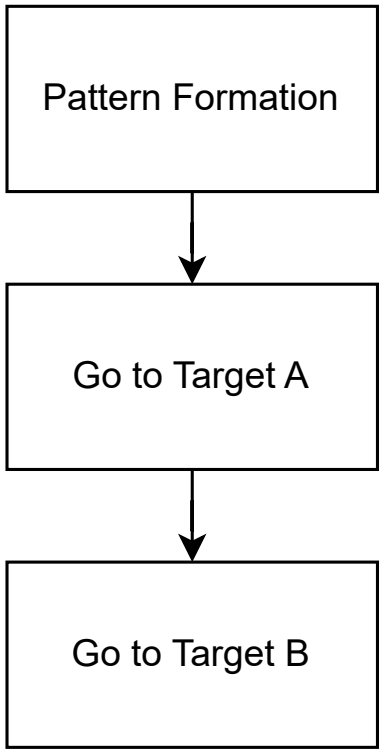
Pattern Formation *Oh et al. 2017*

```
line(...)  || vShape(....)  || centeredCircle(...)
```

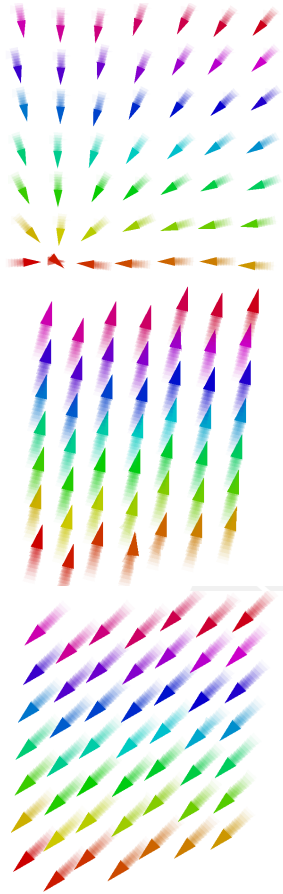


API Overview: Swarm Planning

Expressing a series of plans that change over time that let the swarm achieving different goals

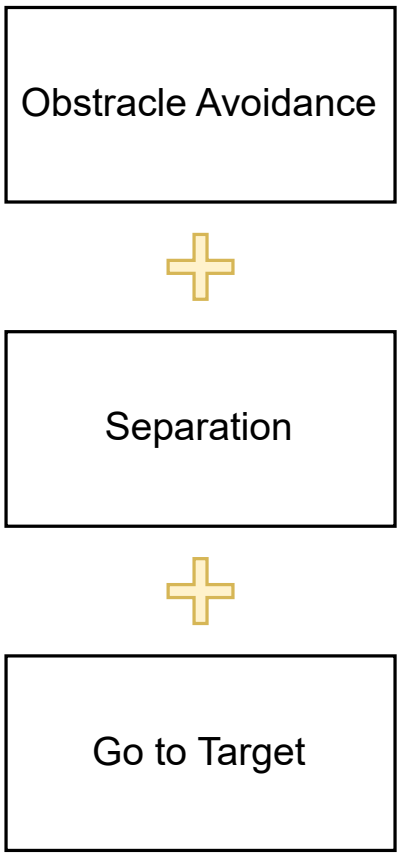



```
execute.once(
  plan {
    sinkAt(leader)
  }.endWhen {
    isTeamFormed(...)
  },
  plan(goToTop).endWhen {
    broadcast(leader, isClose(targetA))
  },
  plan(goToBottom).endWhen {
    broadcast(leader, isClose(targetB))
  }
)
```

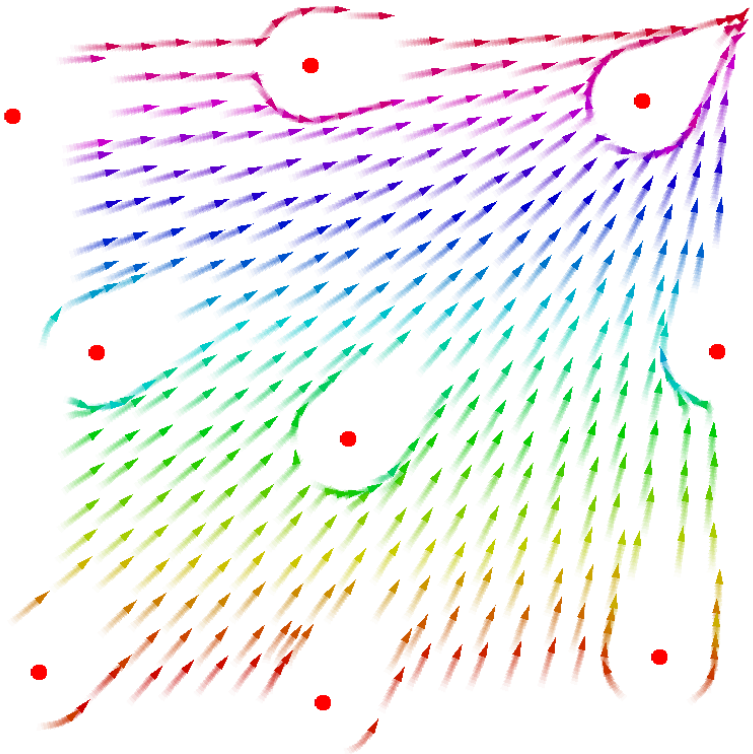


API Overview: Composition

The behaviours can be composed together to create more complex behaviours




```
val avoid = obstacleAvoidance(...)
val separation = separation(...)
val moveTo = goTo(target)
avoid + separation + moveTo
```



Use case: Rescue and Exploration

High-level description

- A swarm of robots is deployed in a rectangular area in which there are injured target
- Two robots type: *healers* and *explorers*
- **Goal**: heal all the injured robots
- `https://github.com/AggregateComputing/experiment-2023-coordination-swarm-behaviour`

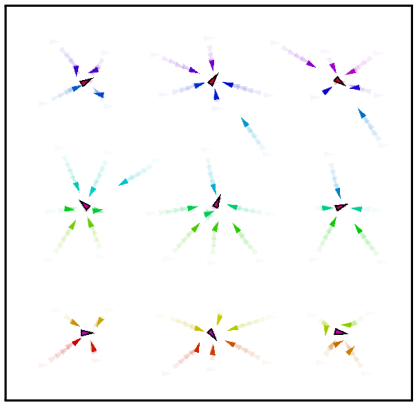
Pseudocode

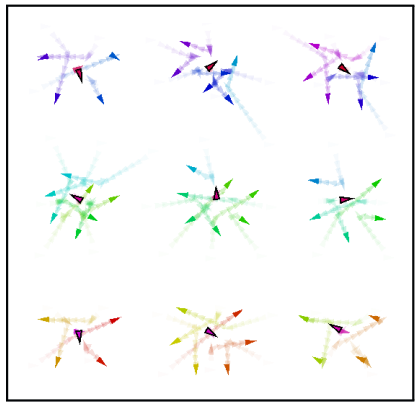
1. Form teams of robots with healers and explorers (*team formation*)
2. Explore the area (*exploration*)
3. When a robot finds an injured, it broadcasts a message (*leader-based*)
4. The healers move towards the injured

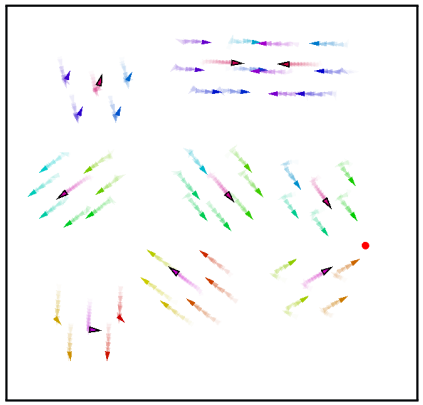
MacroSwarm implementation

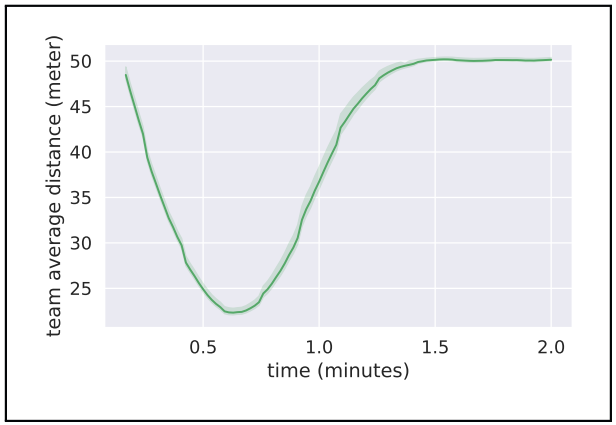
```
execute.repeat(
  plan(formation(lead))
    .endWhen(circleIsFormed),
  plan(wanderInFormation(lead))
    .endWhen(dangerFound),
  plan(goToHealInFormation(lead, inDanger))
      .endWhen(dangerReached),
  plan(heal(healerId, inDanger))
)
```

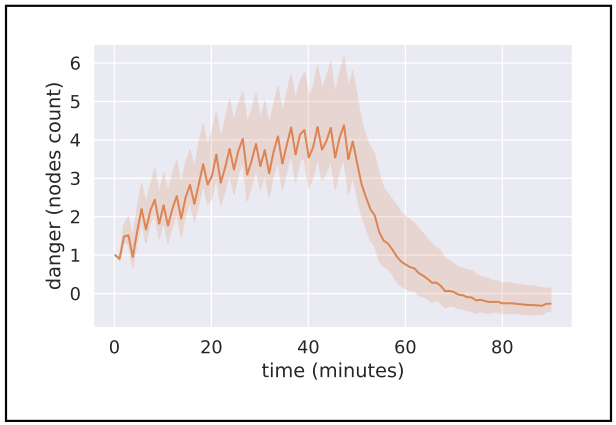
Use case: Results

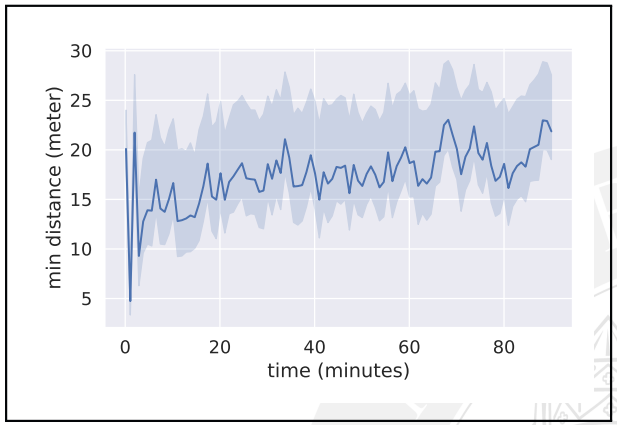












- The swarm is able to form teams with the right shape (left figure)
- The swarm eventually solve the task (center figure)
- They avoid collisions (right figure)

Conclusion

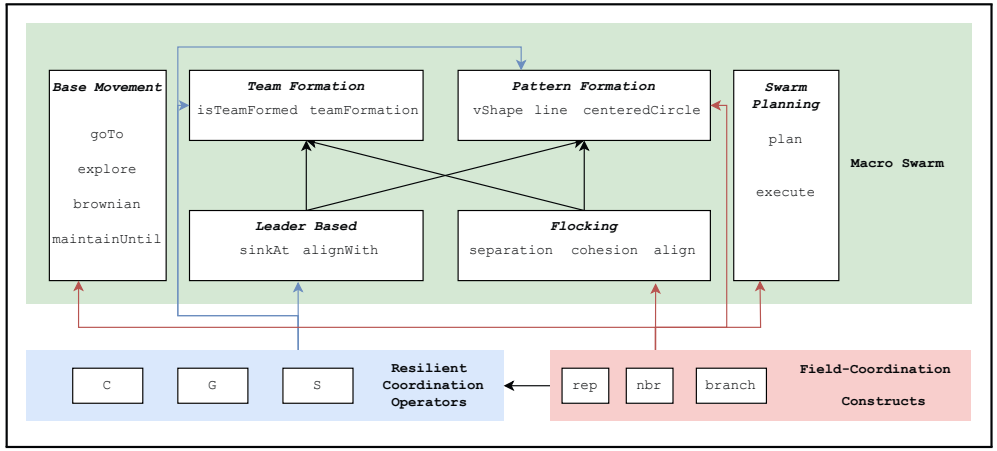
Wrap up

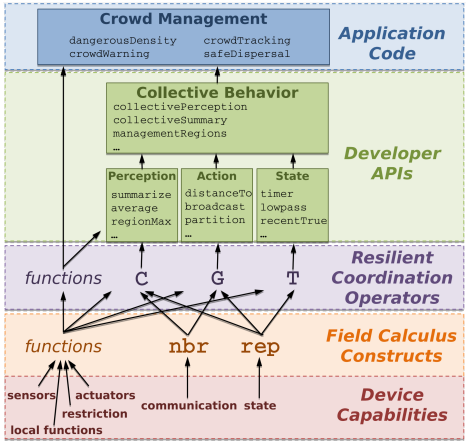
- MacroSwarm: a new programming API for swarm robotics
- API overview covering main swarm behaviours
- Use case to exemplify the API usage

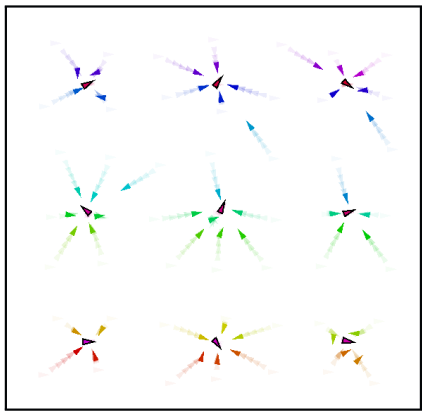
Future work

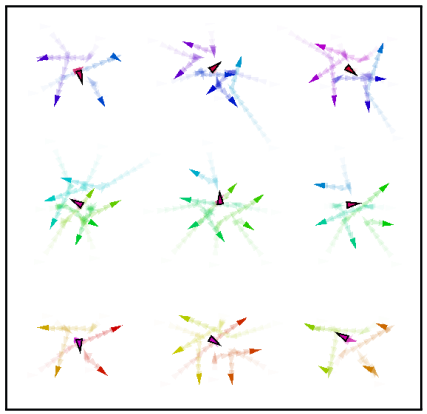
- Integration in a robust swarm robotics framework (e.g., ARGoS, Gazebo)
- Use in a real swarm of robots (e.g., using Crazyflie 2.0)
- Add more behaviours to the API (e.g., complex structures, easier plan composition)

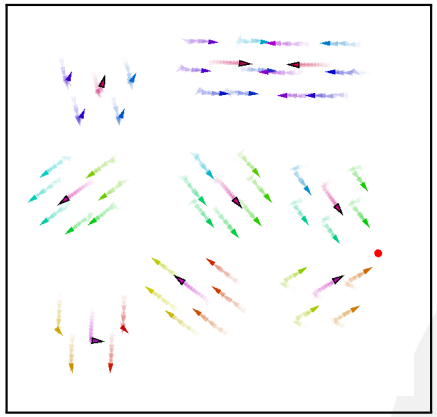
Thank you for your attention!

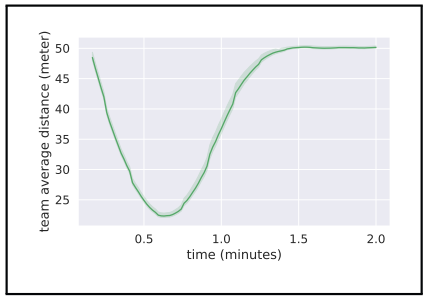


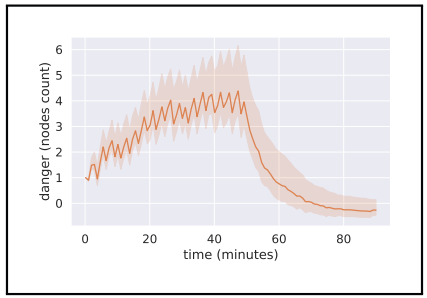


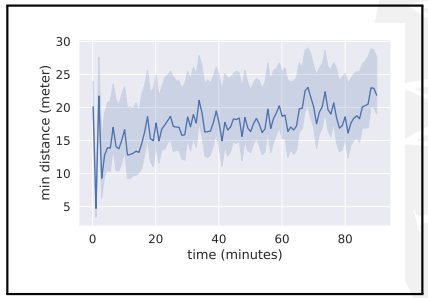










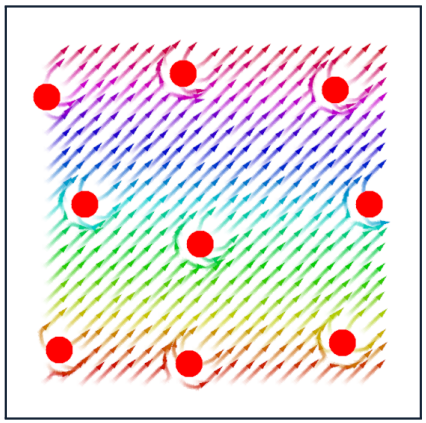
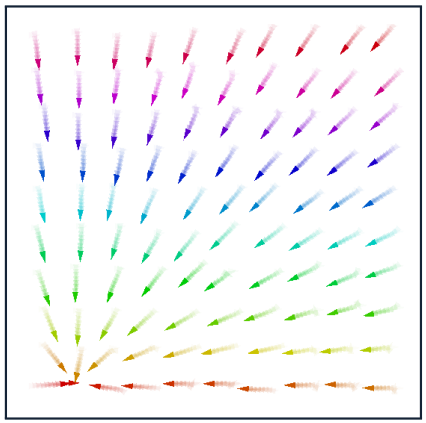
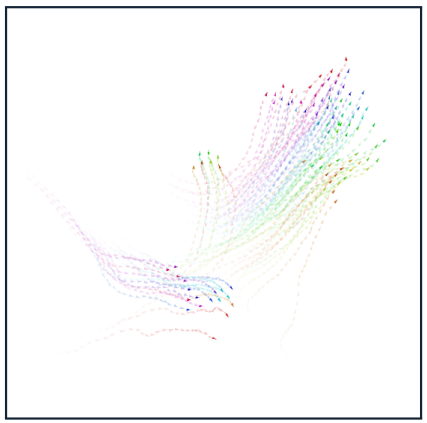
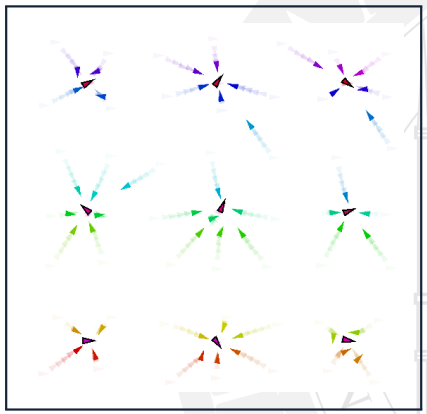


MacroSwarm
Field-based Compositional Framework for Swarm Programming

Gianluca Aguzzi gianluca.aguzzi@unibo.it
Roberto Casadei roby.casadei@unibo.it
Mirko Viroli mirko.viroli@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum – Università di Bologna

Talk @ COORDINATION 2023

References I

[1] Dongbing Gu and Zongyao Wang. "Leader-Follower Flocking: Algorithms and Experiments". In: *IEEE Trans. Control. Syst. Technol.* 17.5 (2009), pp. 1211–1219. doi: 10.1109/TCST.2008.2009461. url: https://doi.org/10.1109/TCST.2008.2009461.

[2] Gregory Mone. "Rise of the swarm". In: *Communications of the ACM* 56.3 (Mar. 2013), pp. 16–17.

[3] Hyondong Oh et al. "Bio-inspired self-organising multi-robot pattern formation: A review". In: *Robotics Auton. Syst.* 91 (2017), pp. 83–100. doi: 10.1016/j.robot.2016.12.006. url: https://doi.org/10.1016/j.robot.2016.12.006.

[4] Craig W Reynolds. "Flocks, herds and schools: A distributed behavioral model". In: *Computer Graphics (Proceedings of ACM SIGGRAPH 87)*. 1987, pp. 25–34.

[5] Melanie Schranz et al. "Swarm robotic behaviors and current applications". In: *Frontiers in Robotics and AI* 7 (2020), p. 36.

[6] Franck Varenne et al. "Programming the emergence in morphogenetically architected complex systems". In: *Acta biotheoretica* 63 (2015), pp. 295–308.

References II

[7] Mirko Viroli et al. "From field-based coordination to aggregate computing". In: *Coordination Models and Languages: 20th*. Springer. 2018, pp. 252–279.