Marcin Biczyski
m.biczyski@cranfield.ac.uk

2017-2018 AVDC MSc Individual Research Project:

Domain Specific Language for Drone Swarm Evasive Manoeuvring

Overview

17th July 2018

Supervisors: Prof Rafał Żbikowski and Dr Al Savvaris

Marcin Biczyski
m.biczyski@cranfield.ac.uk

Presentation outline

dSMOL

Drone Swarm Modelling and Optimisation Language

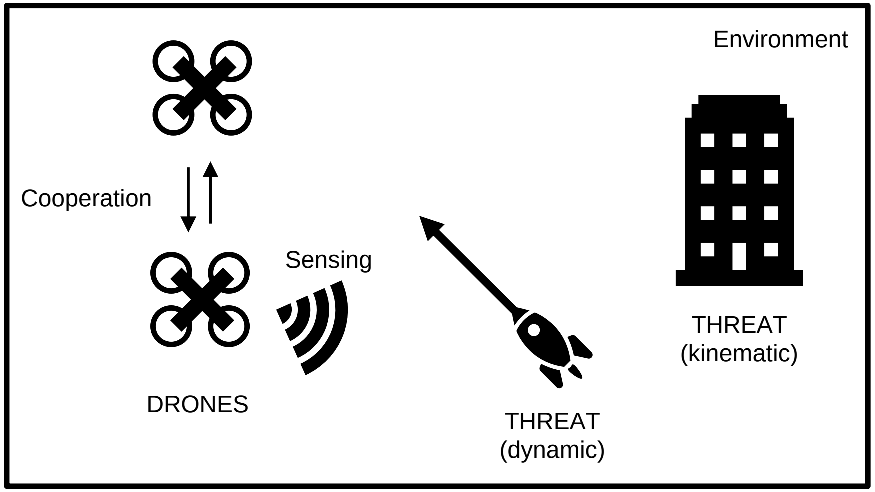
1. Project objectives
2. Methodology
3. Overall progress
4. Current features
   a) Why Racket?
5. Sample project
6. Conclusions
7. Next steps and demo project

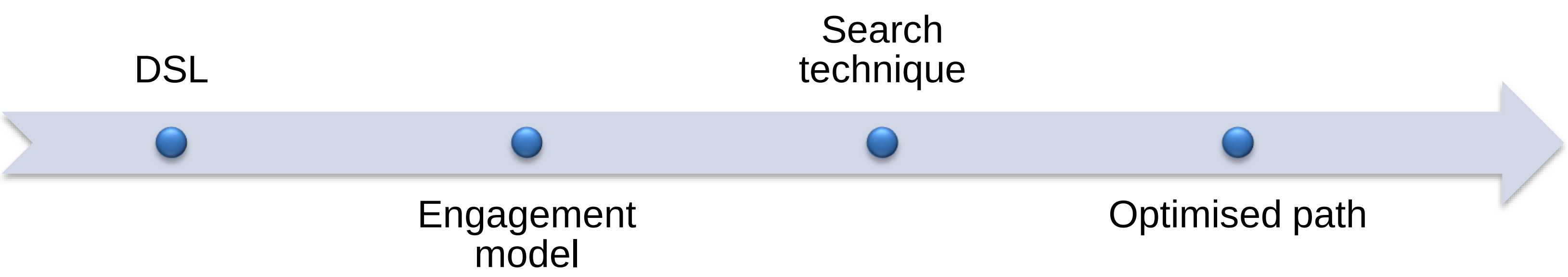
Source: http://www.tandemnsi.com/2017/01/darpa-wants-help-building-swarm-drone-test-bed/

Marcin Biczyski
m.biczyski@cranfield.ac.uk

Project objectives

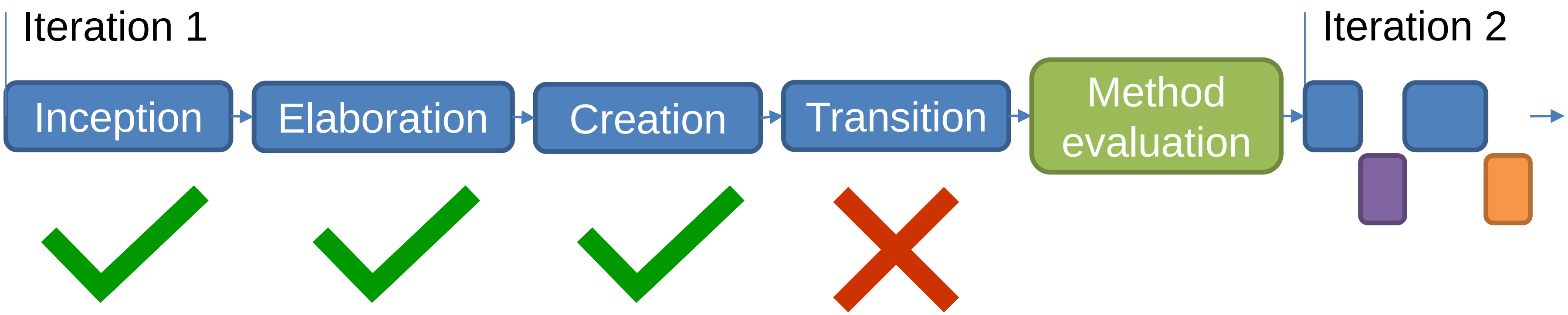
1. Create a Domain Specific Language (DSL) capable of:
   a. Modelling evasive manoeuvres
   b. Decentralised decision making algorithms implementation
   c. Imposing different types of constraints
   d. Drone path generation
2. Implement standard search techniques (e.g. simulated annealing)
3. Optimise drone paths in the presence of threats and obstacles





Marcin Biczyski
m.biczyski@cranfield.ac.uk

Methodology evaluation



Transition

- ✗ T1 – syntactic quality
- ✓ T3 – pragmatics evaluation
  - ✓ T3.1 – comprehension, creation, and modification
  - ✓ T3.2 – support for use cases
- ✓ T4 – organisational evaluation

- ✗ T2 – semantics evaluation
  - ✗ T2.1 – feature model satisfiability
  - ✗ T2.2 – relevant API usages
  - ✓ T2.3 – mapping definitions
  - ✗ T2.4 – implementation of code queries and transformations

Marcin Biczyski
m.biczyski@cranfield.ac.uk

Methodology update

Framework for Qualitative Assessment of DSLs

- Functional stability
- Usability
- Reliability
- Maintainability
- Productivity
- Extensibility
- Compatibility
- Expressiveness
- Reusability
- Integrability

Priority
- Mandatory
- Desirable
- Nice to have

Support
- No support
- Some support
- Strong support
- Full support

Success level
- Incomplete
- Satisfactory
- Effective

Kahraman, G. and Bilgen, S. (2015) ‘A framework for qualitative assessment of domain-specific languages’, *Software & Systems Modeling*. Springer Berlin Heidelberg, 14(4), pp. 1505–1526. doi: 10.1007/s10270-013-0387-8.

Centre for Autonomous and Cyber-Physical Systems
School of Aerospace, Transport and Manufacturing

Marcin Biczyski
m.biczyski@cranfield.ac.uk

Progress - value & use cases

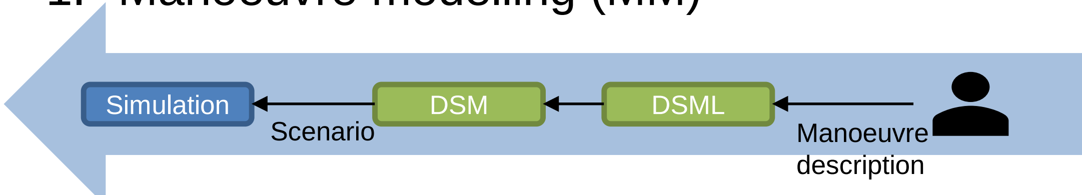
Added value

1. Provide drone swarm path generation utility
2. Enable modelling of threats and obstacles
3. Ease the process of simulation and optimisation

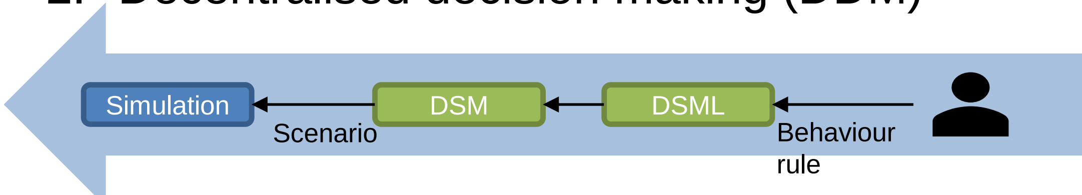
Able to simulate any type of vehicle, not only flying.

Use cases

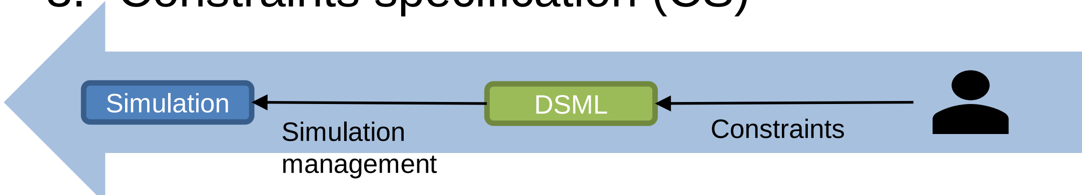
1. Manoeuvre modelling (MM)



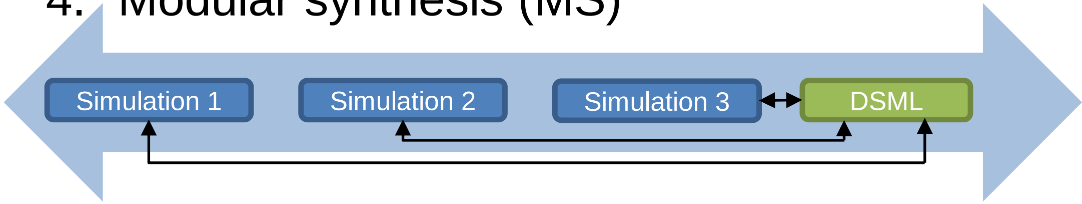
2. Decentralised decision making (DDM)



3. Constraints specification (CS)

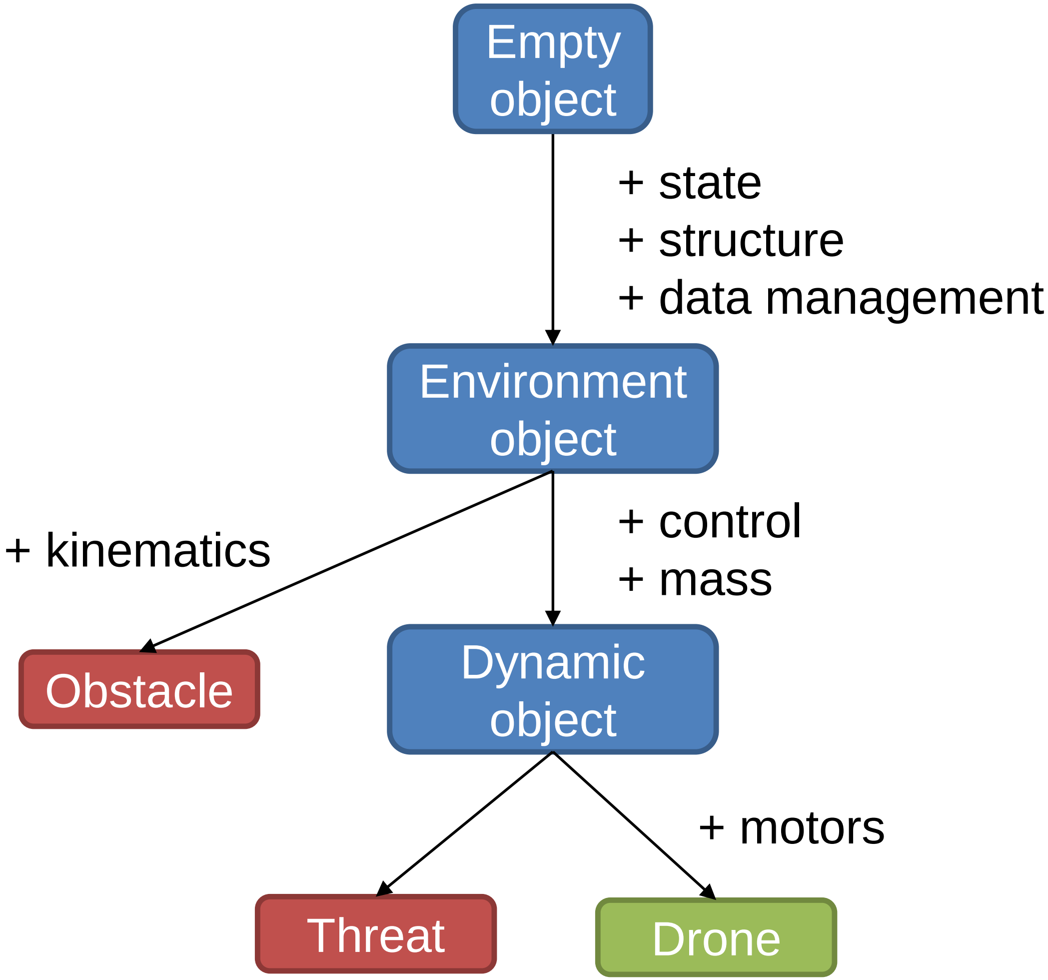


4. Modular synthesis (MS)



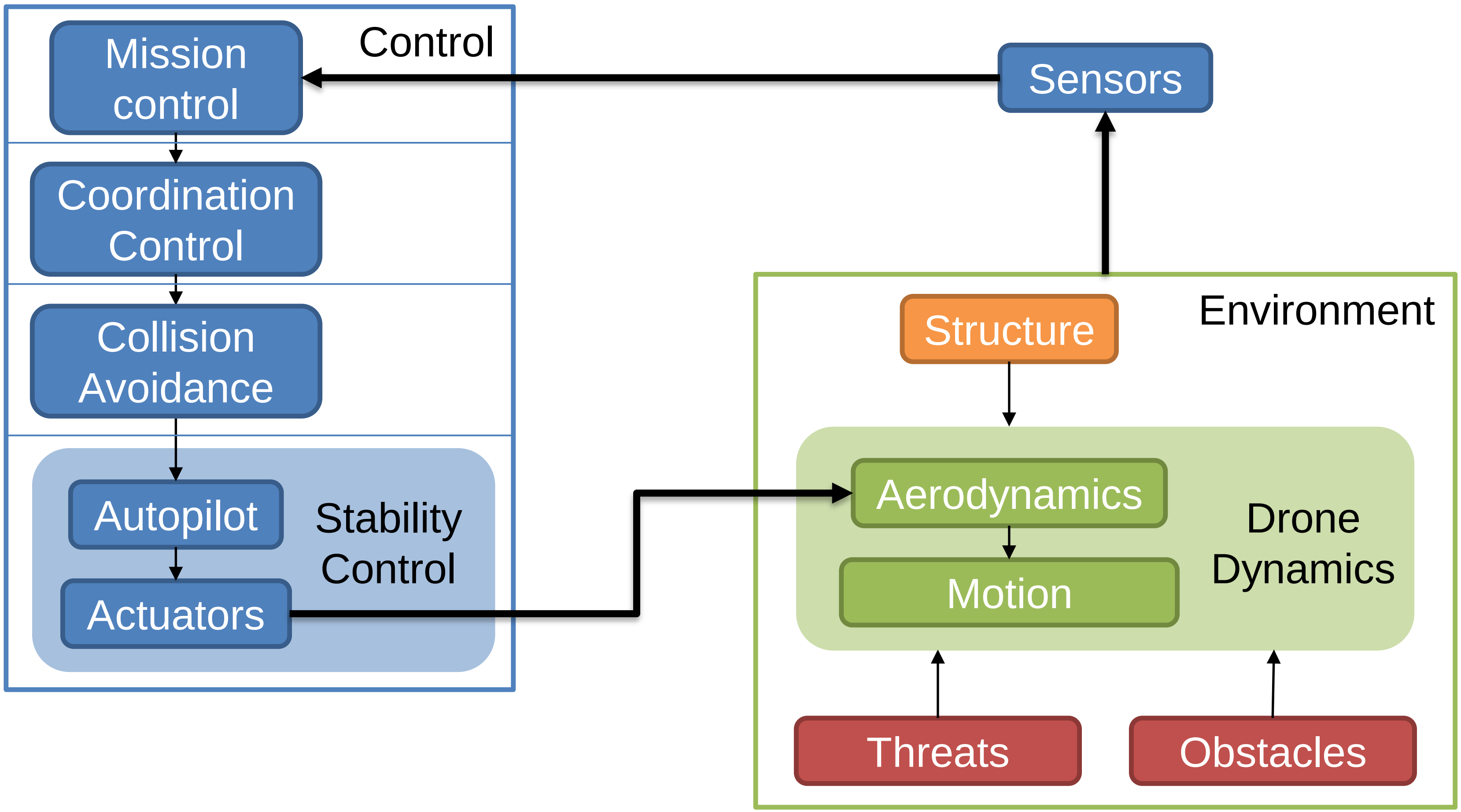
Marcin Biczyski
m.biczyski@cranfield.ac.uk

Features – types of objects



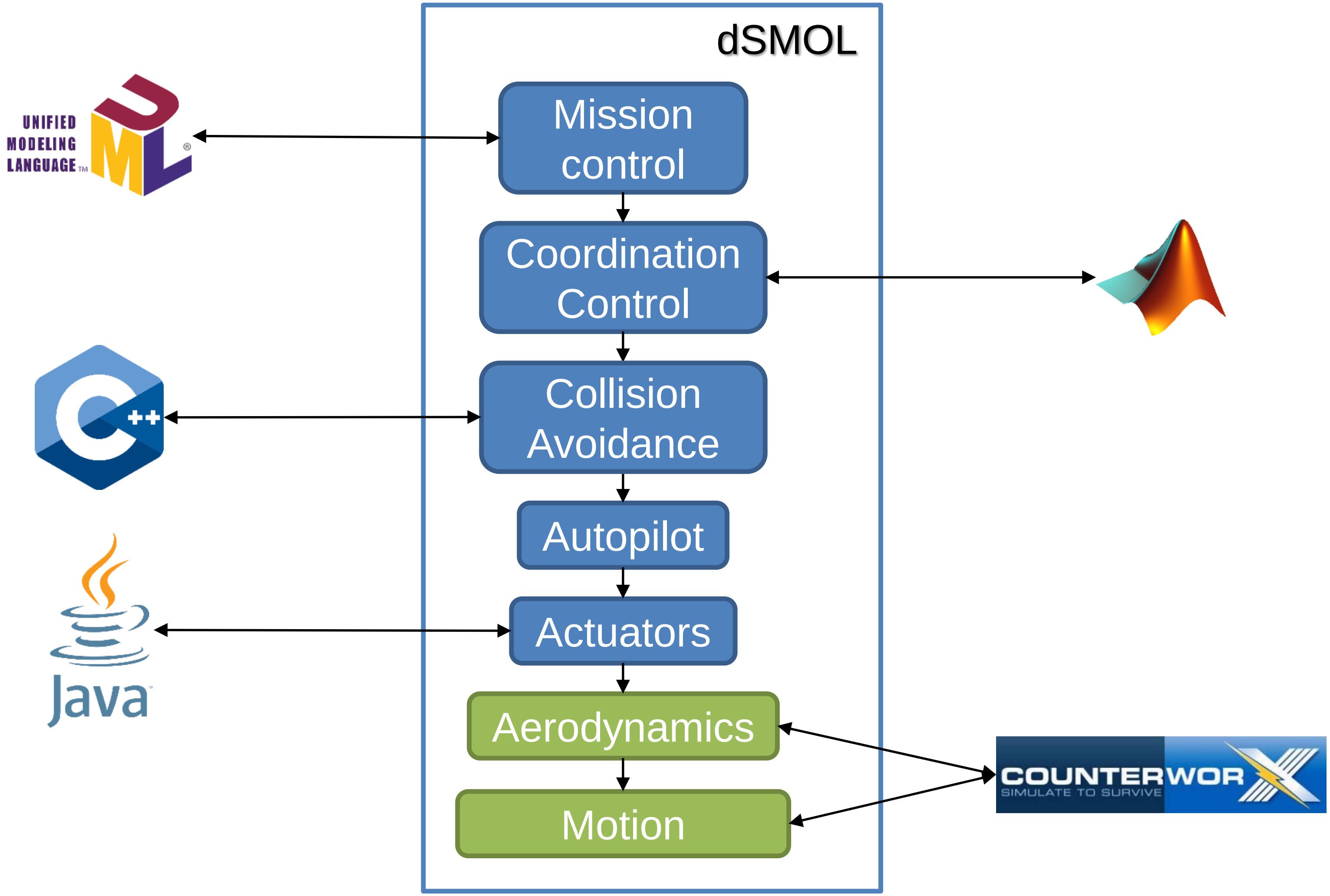
Marcin Biczyski
m.biczyski@cranfield.ac.uk

Features – layered control



Marcin Biczyski
m.biczyski@cranfield.ac.uk

Features - modularity



Marcin Biczyski
m.biczyski@cranfield.ac.uk

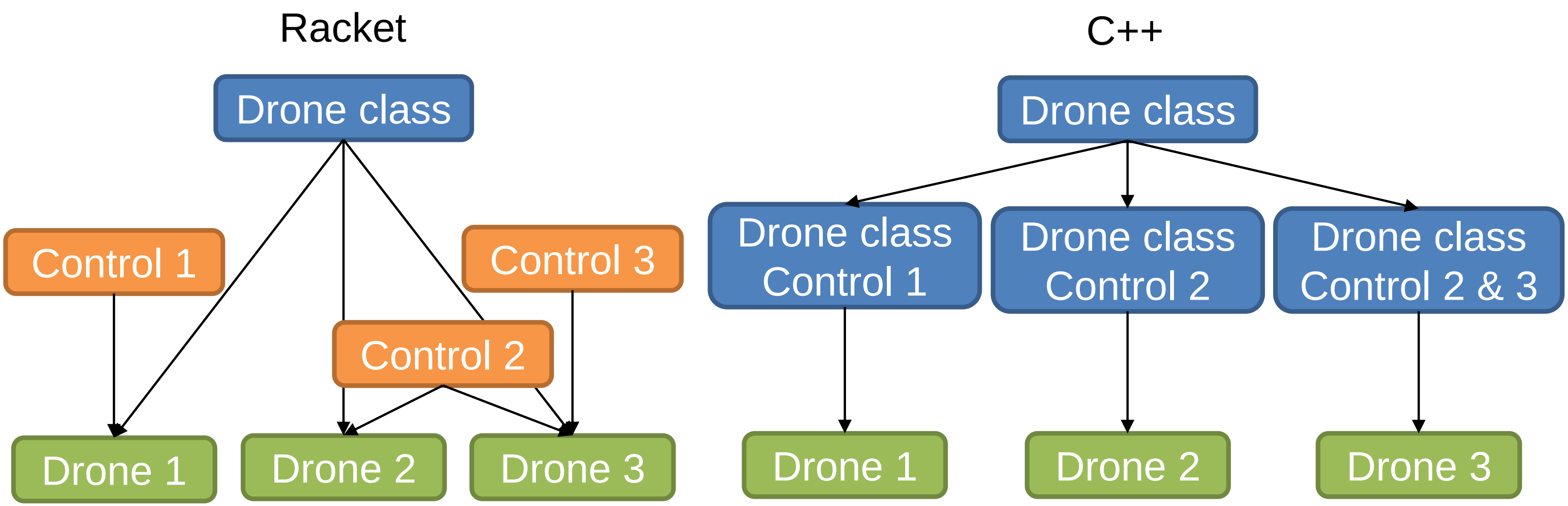
Why Racket?

Racket is a general-purpose programming language as well as the world's first ecosystem for developing and deploying new languages.

1. *Racket is about creating new programming languages quickly.*
2. *Racket provides building blocks for strong protection mechanisms.*
3. *Racket turns extra-linguistic mechanisms into linguistic constructs.*

Source: Felleisen et al. 'The Racket Manifesto'

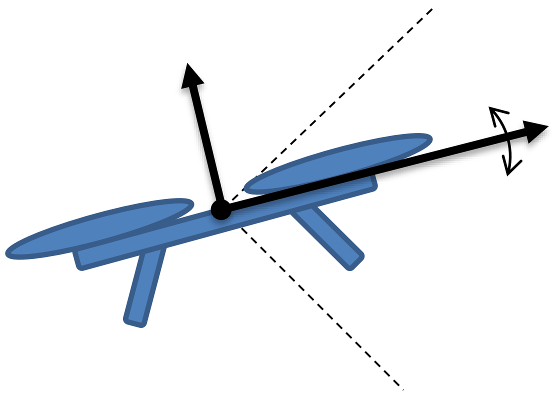
Most useful feature – metaobject system



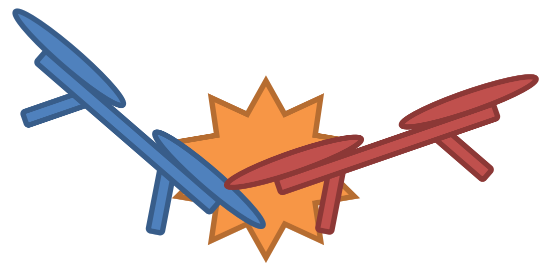
Marcin Biczyski
m.biczyski@cranfield.ac.uk

Features - constraints

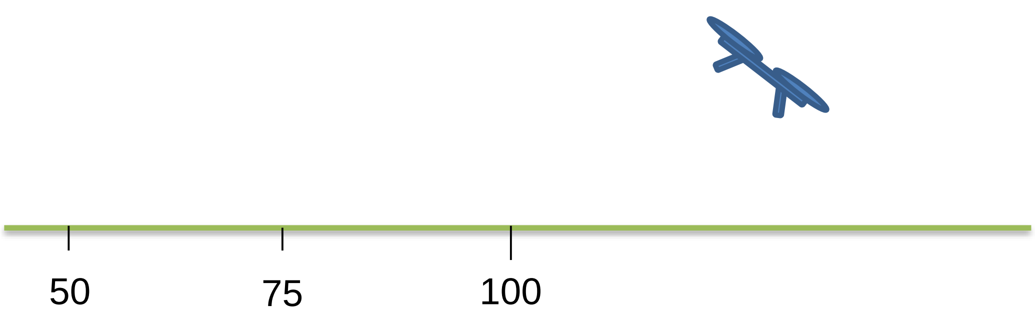
Motion constraints



Interaction constraints



Simulation area constraints



Marcin Biczyski
m.biczyski@cranfield.ac.uk

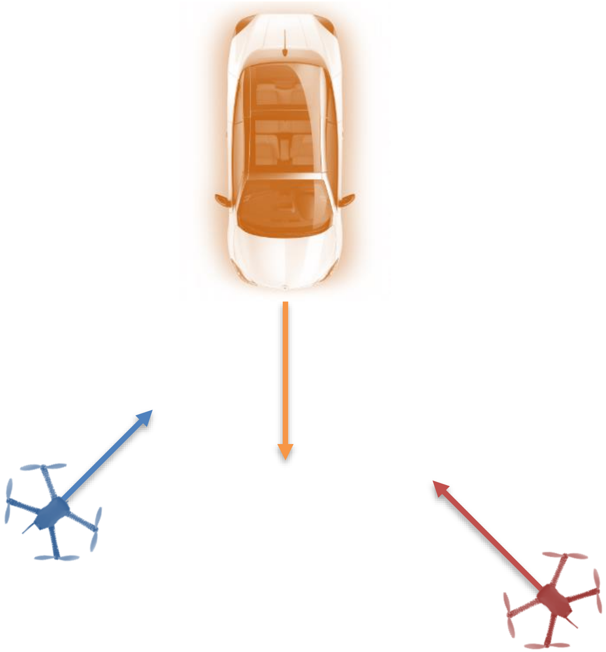
Sample project – overview

Drone 1 (blue):
- Desired velocity: 3 m/s
- Desired heading: 45°
- Desired altitude: 1.5 m
- Dynamic model in Matlab
- Autopilot in dSMOL

Drone 2 (red):
- Desired velocity: 5 m/s
- Desired heading: -45°
- Desired altitude: 1.5 m
- Dynamic model in Matlab
- Autopilot in dSMOL

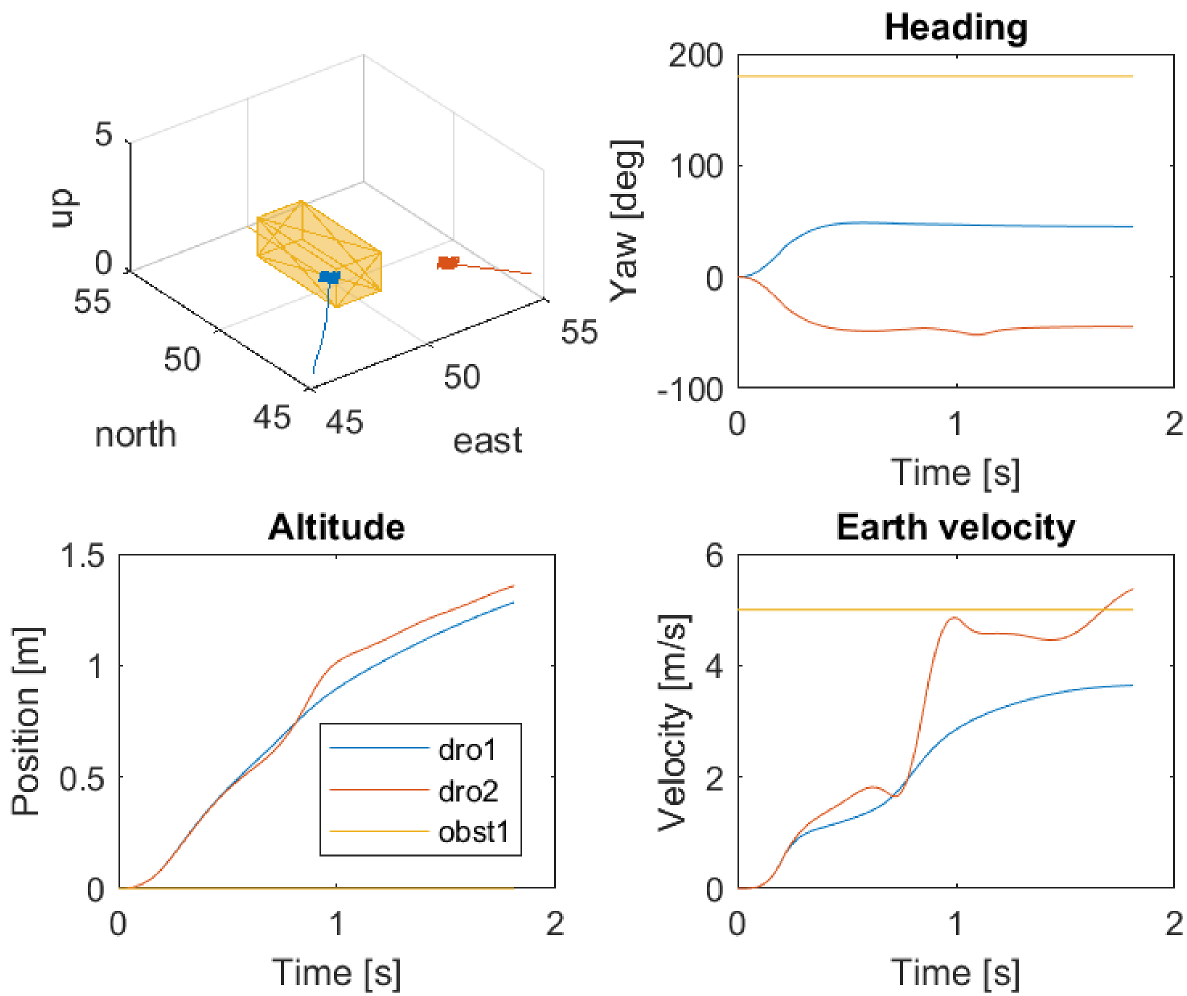
Initial conditions:
- Velocity: 0 m/s
- Heading: 0°
- Altitude: 0 m
- Structure: IRIS+

Car 1 (yellow):
- Velocity: 5 m/s
- Heading: 180°
- Altitude: 0 m
- Kinematic model in dSMOL
- Structure: Renault Megane RS



Marcin Biczyski
m.biczyski@cranfield.ac.uk

Sample project – simulation



Marcin Biczyski
m.biczyski@cranfield.ac.uk

Conclusions

Strengths

- Layered control
- Openness to modification
- Exhausts the domain

Weaknesses

- Simple constraints
- No sensor implementation
- No explicit evasive manoeuvres

Opportunities

- Modularity
- Generalizability
- Control of every aspect of simulation

Threats

- Niche language
- No error-proofing
- Long simulation time

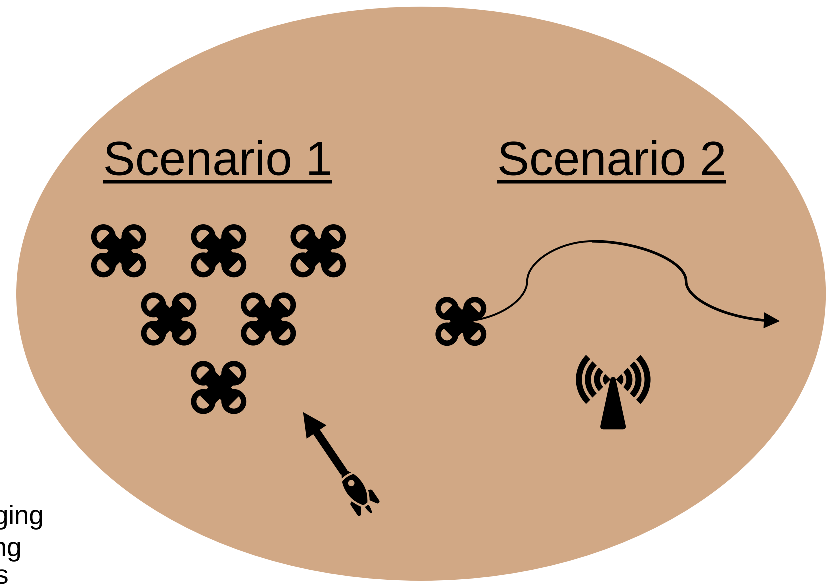
Marcin Biczyski
m.biczyski@cranfield.ac.uk

Next steps

Iteration 3

- Fitness measurement
- Selective parameterization
- Optimization

Scenario building



Iteration 4

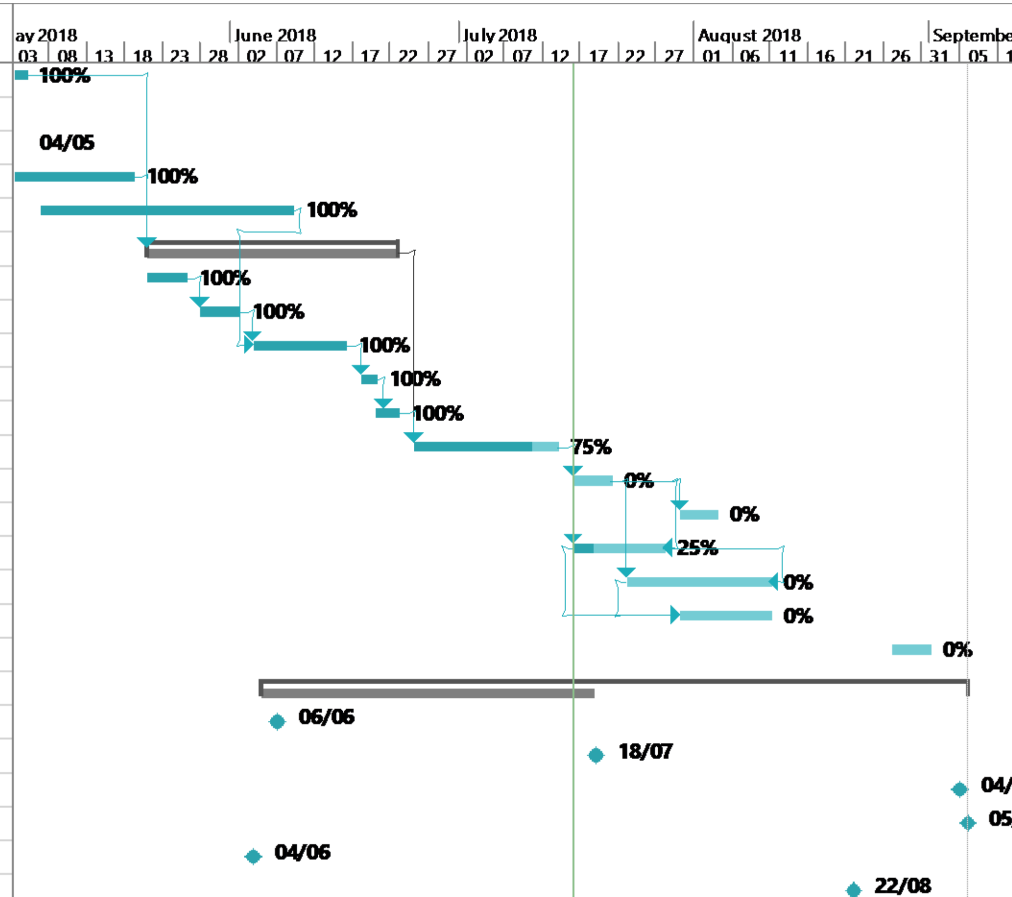
- Debugging
- Finishing touches
- Results presentation

Deployment

Marcin Biczyski
m.biczyski@cranfield.ac.uk

Gantt chart

ID		Task Mode	Task Name	Duration	Start	Finish
1	✓		Scope and focus	41 days	Fri 09/03/18	Fri 04/05/18
2	✓		Initial meeting	0 days	Fri 09/03/18	Fri 09/03/18
3	✓		Industry meeting	0 days	Fri 04/05/18	Fri 04/05/18
4	✓		Literature review	51 days	Fri 09/03/18	Fri 18/05/18
5	✓		LISP tutorial	25 days	Mon 07/05/18	Fri 08/06/18
6	✓		**Iteration 1**	**25 days**	**Mon 21/05/18**	**Fri 22/06/18**
7	✓		Inception	5 days	Mon 21/05/18	Fri 25/05/18
8	✓		Elaboration	5 days	Mon 28/05/18	Fri 01/06/18
9	✓		Creation	10 days	Mon 04/06/18	Fri 15/06/18
10	✓		Transition	2 days	Mon 18/06/18	Tue 19/06/18
11	✓		Method evaluati	3 days	Wed 20/06/18	Fri 22/06/18
12			Iteration 2	15 days	Mon 25/06/18	Fri 13/07/18
13			Iteration 3	5 days	Mon 16/07/18	Fri 20/07/18
14			Iteration 4	5 days	Mon 30/07/18	Fri 03/08/18
15			Demo project	10 days	Mon 16/07/18	Fri 27/07/18
16			Thesis writing	15 days	Mon 23/07/18	Fri 10/08/18
17			Thesis feedback	10 days	Mon 30/07/18	Fri 10/08/18
18			Viva and Poster pre	5 days	Mon 27/08/18	Fri 31/08/18
19			**Evaluation**	**67 days**	**Tue 05/06/18**	**Wed 05/09/18**
20	✓		1st Interim prese	2 days	Tue 05/06/18	Wed 06/06/18
21			2nd Interim pres	2 days	Tue 17/07/18	Wed 18/07/18
22			Oral presentatio	2 days	Mon 03/09/18	Tue 04/09/18
23			Poster presentat	1 day	Wed 05/09/18	Wed 05/09/18
24	✓		Previous works rev	0 days	Mon 04/06/18	Mon 04/06/18
25			Thesis hand in	0 days	Wed 22/08/18	Wed 22/08/18



Marcin Biczyski
m.biczyski@cranfield.ac.uk

Risk register

Project Number	Project name	Date
IRP 9	Domain Soecific Language for Drone Swarm Evasive manoeuvring	13/07/2018

Student	End user	Project type
Marcin Biczyski	Programmer - simulation specialist	Small

Project supervisor	Project sponsor
Prof. Rafał Żbikowski	MASS

Number	Status	Category	Risk event	Probability	Impact	Priority	Risk response	Owner
1	Closed	Project management	DSL is developed in a language new to the student. If the programming ability is not sufficient at the start of implementation, other activities may be delayed.	Medium	Medium	Medium	Avoidance - consultations with experts, study based on well-respected sources	Marcin Biczyski
2	Open	Implementation	The first iteration is designed to be very basic. If the work involved is too high, not all proposed features may get implemented in future iterations, making the final DSL not versatile enough for general use.	Low	High	Medium	Mitigation - components' priority analysis, modular approach - outsourcing components	Marcin Biczyski
3	Open	Security	LISP allows for free access to the source code. If not enough care is taken, the DSL may become prone to user errors	Medium	Low	Low	Acceptance - feature not necessary for achieving performance	Marcin Biczyski
4	Open	Delivery	The demo project is complicated. There may not be enough ready-made components or a too much changes may be needed to be able to finish it in time.	Medium	Medium	Medium	Mitigation - Simplification of demo project based on available resources	Marcin Biczyski
5	Open	Delivery	Due to modular nature of the system, invoking specific modules may require a lot of time durning runtime thus slowing down the simulation process. This may result in the optimisation process taking too much time.	High	Medium	Medium	Acceptance - only the proof that the process works as expected is required, not the complete solution	Marcin Biczyski

Marcin Biczyski
m.biczyski@cranfield.ac.uk

Thank you!