From AgentSpeak to C
for Unmanned Aerial Vehicles

Samuel Bucheli[a]

Department of Computer Science
University of Oxford

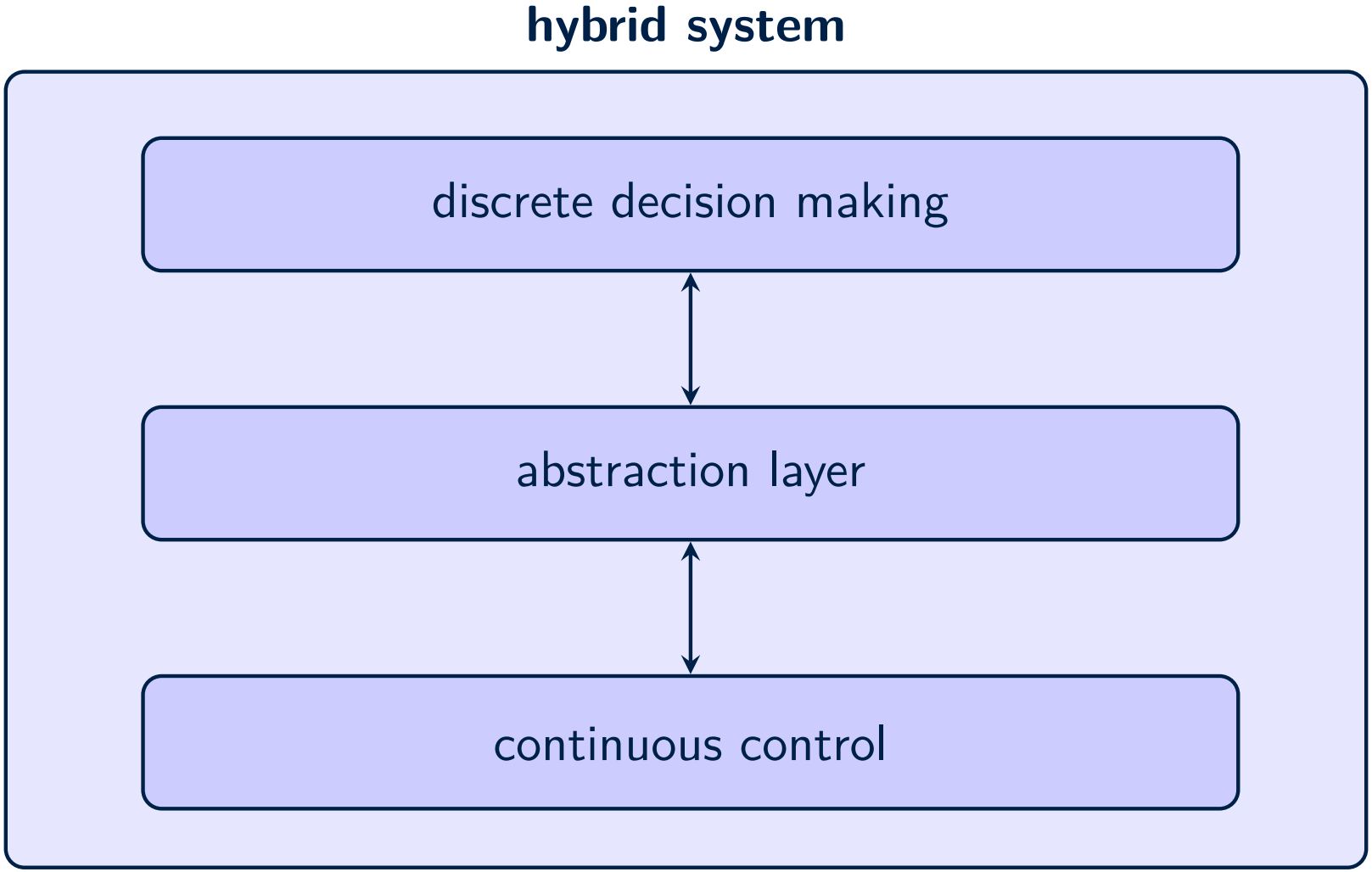
May 28, 2015

[a] joint work with Daniel, Ruben, and Ashutosh

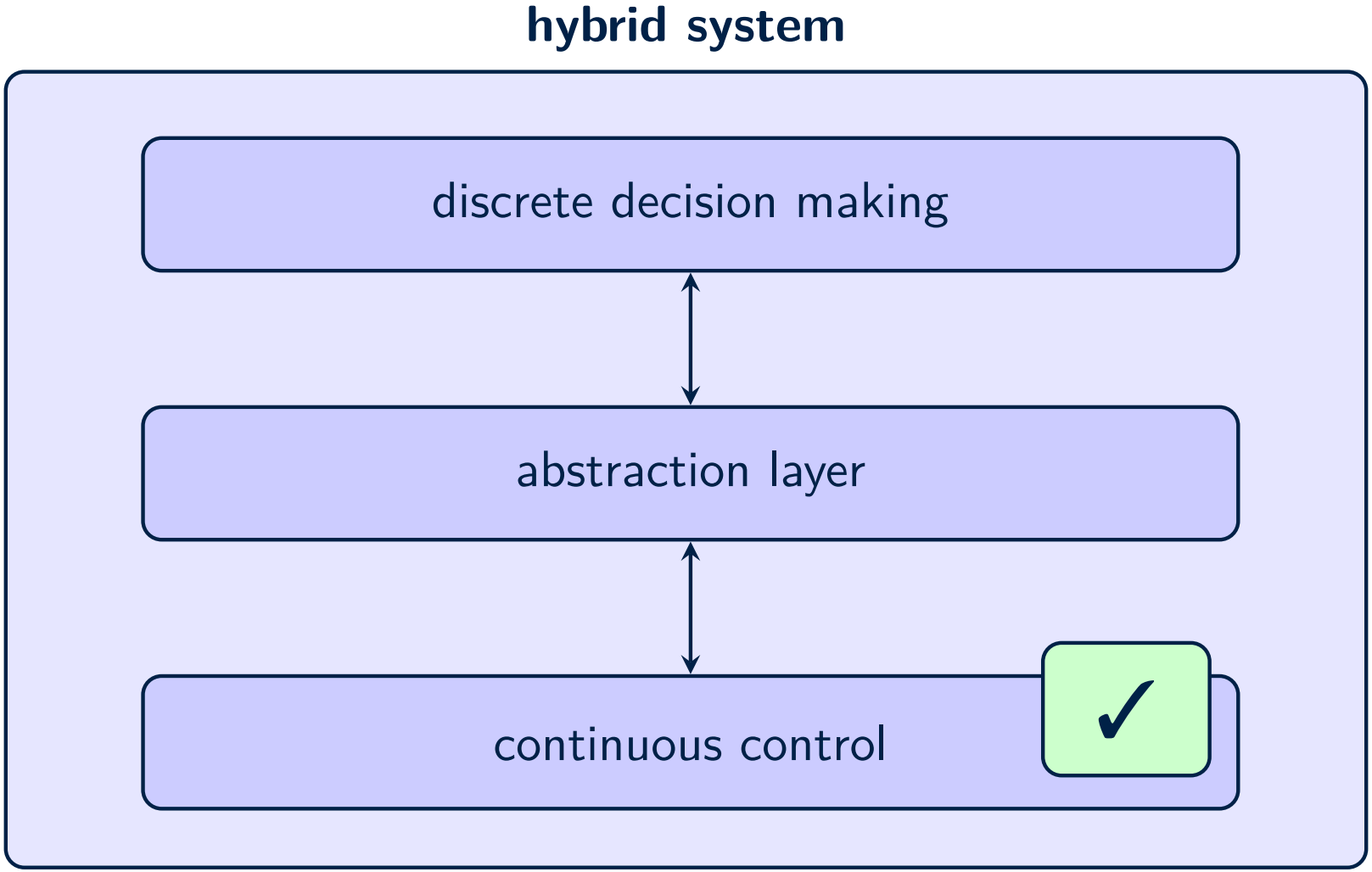
In this Talk

- autonomous agents as hybrid systems,
- safety considerations: DO-178C and MISRA C,
- AgentSpeak as a modeling language for high-level behavior,
- from AgentSpeak to C: examples and experiments,
- further work: translation validation.

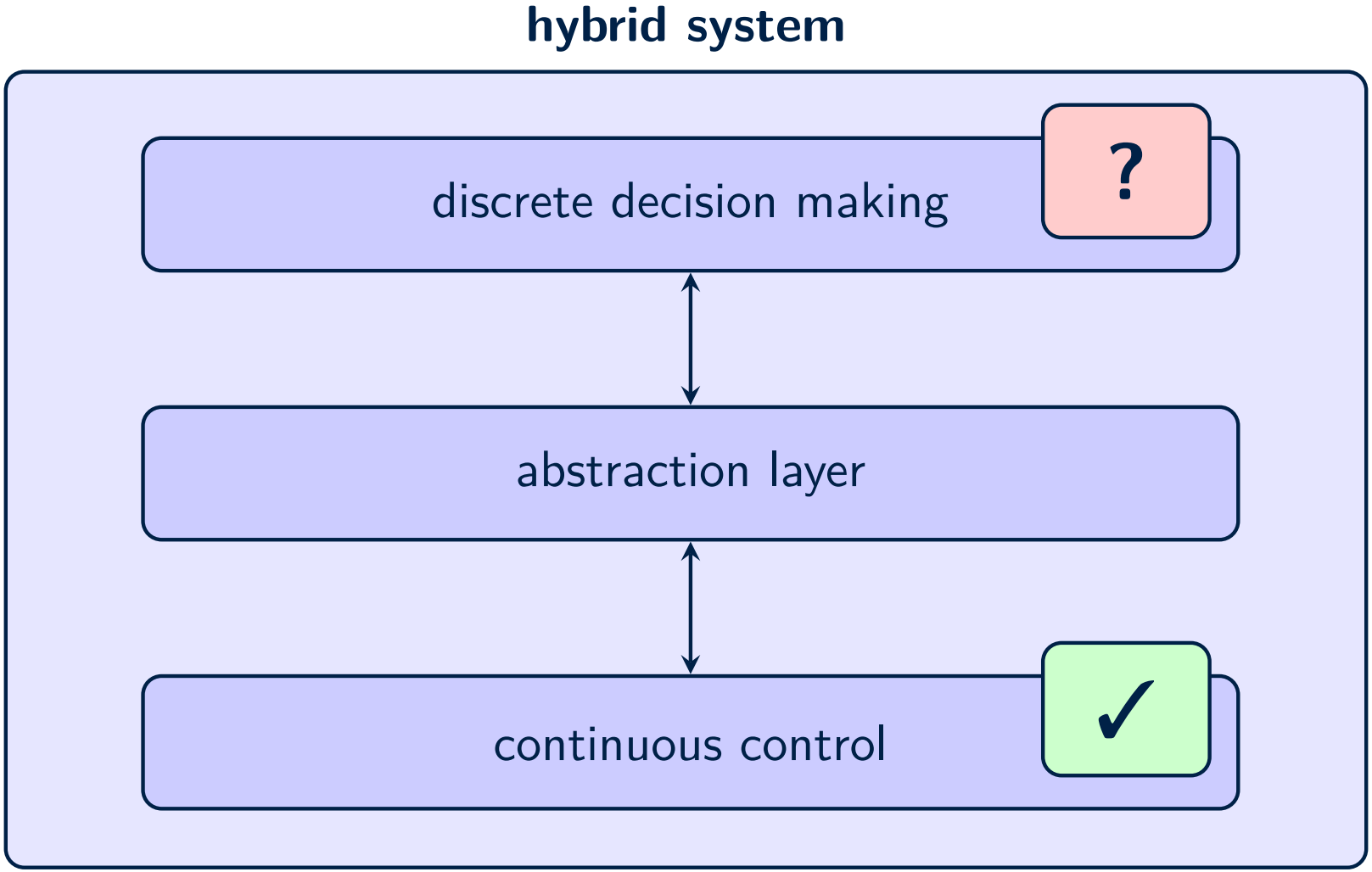
Autonomous Agents



Autonomous Agents



Autonomous Agents



DO-178C

Software Considerations in Airborne Systems and Equipment Certification

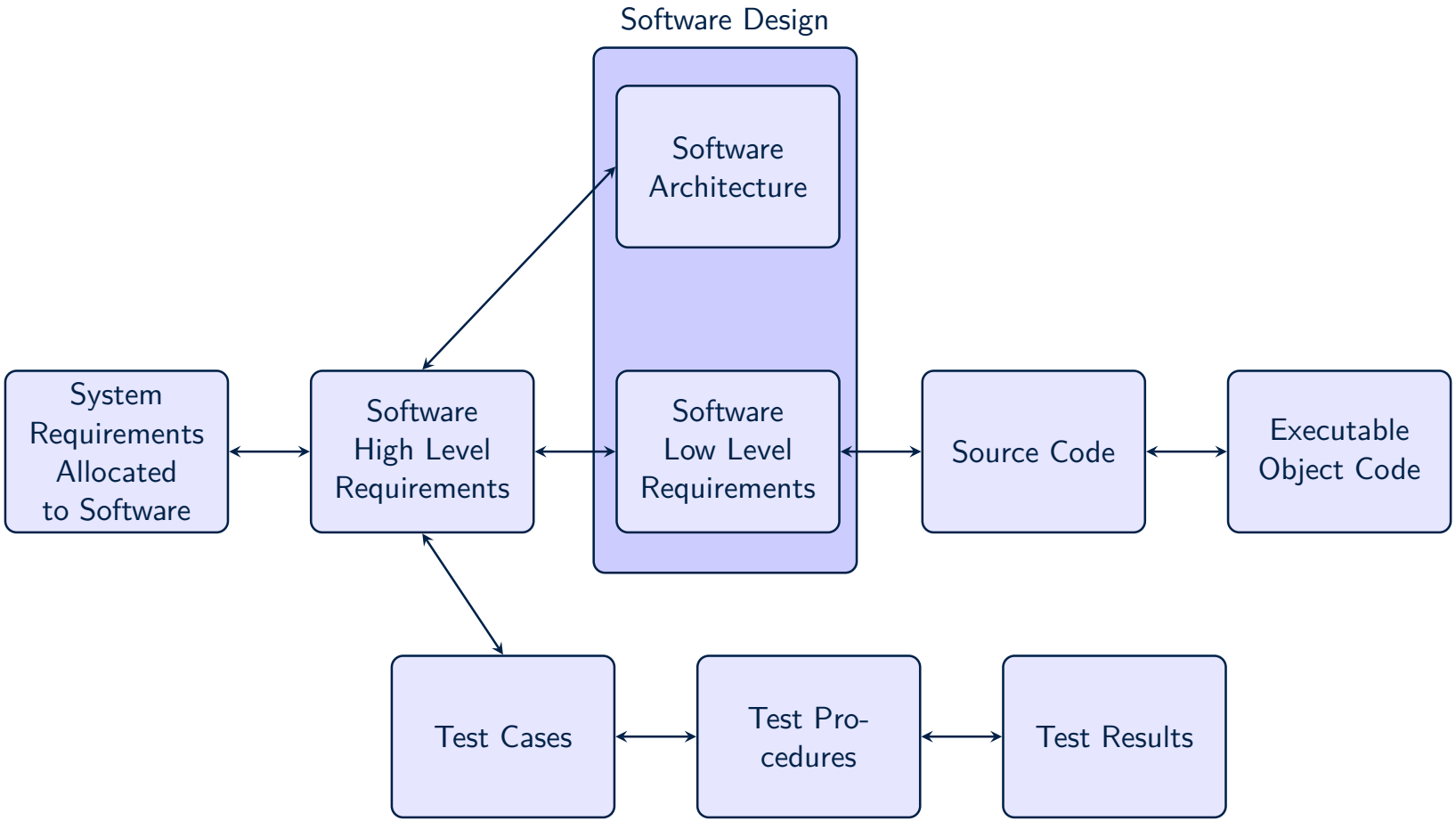


image based on `http://en.wikipedia.org/wiki/DO-178C`

MISRA C

Motor Industry Software Reliability Association

...

16.2 (req) Functions shall not call themselves, either directly or indirectly.

...

20.4 (req) Dynamic heap memory allocation shall not be used.

...

The Autopilot

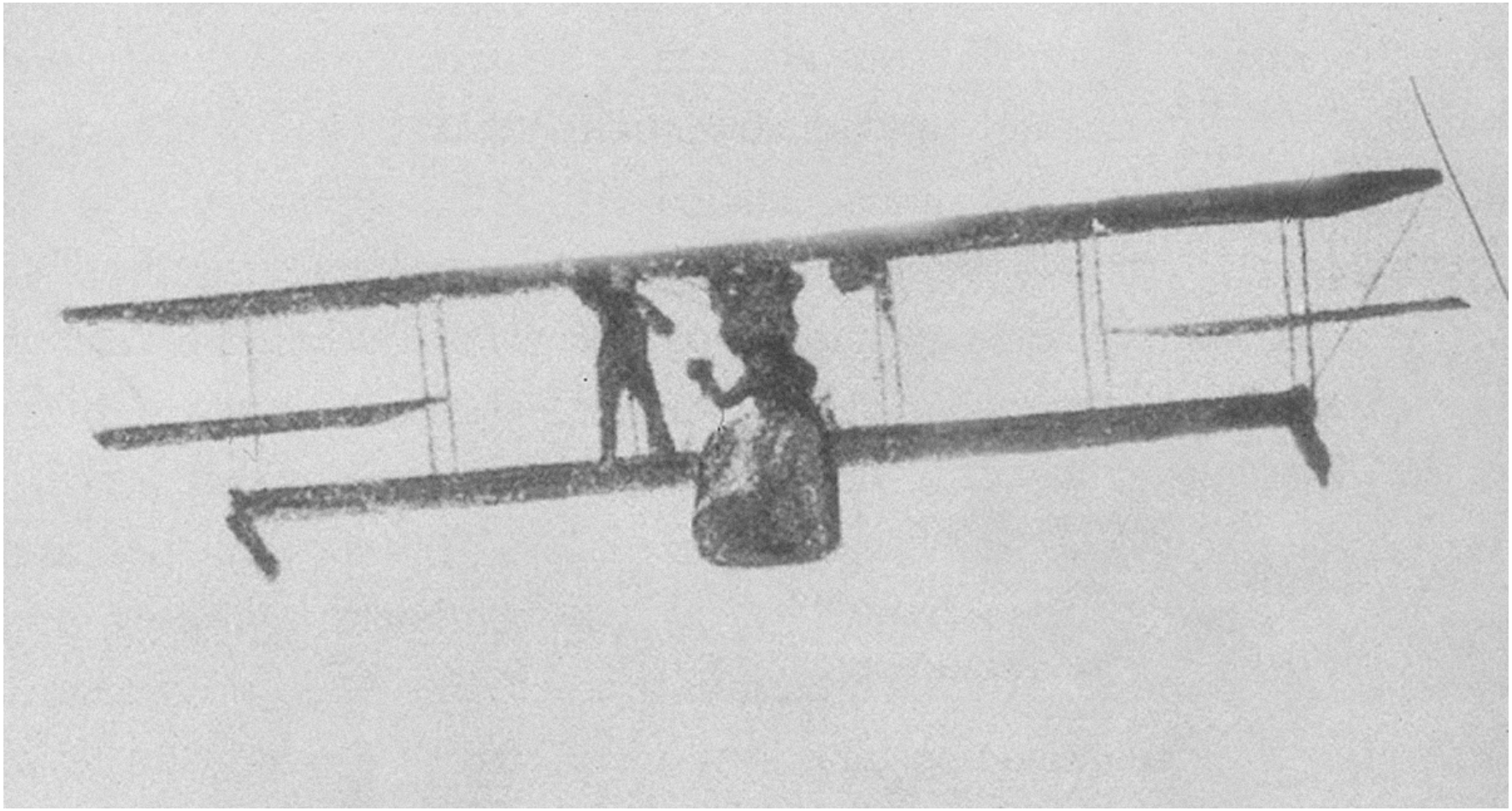


image from http://fly.historicwings.com/2012/08/george-the-autopilot/

Sample Flight Plan

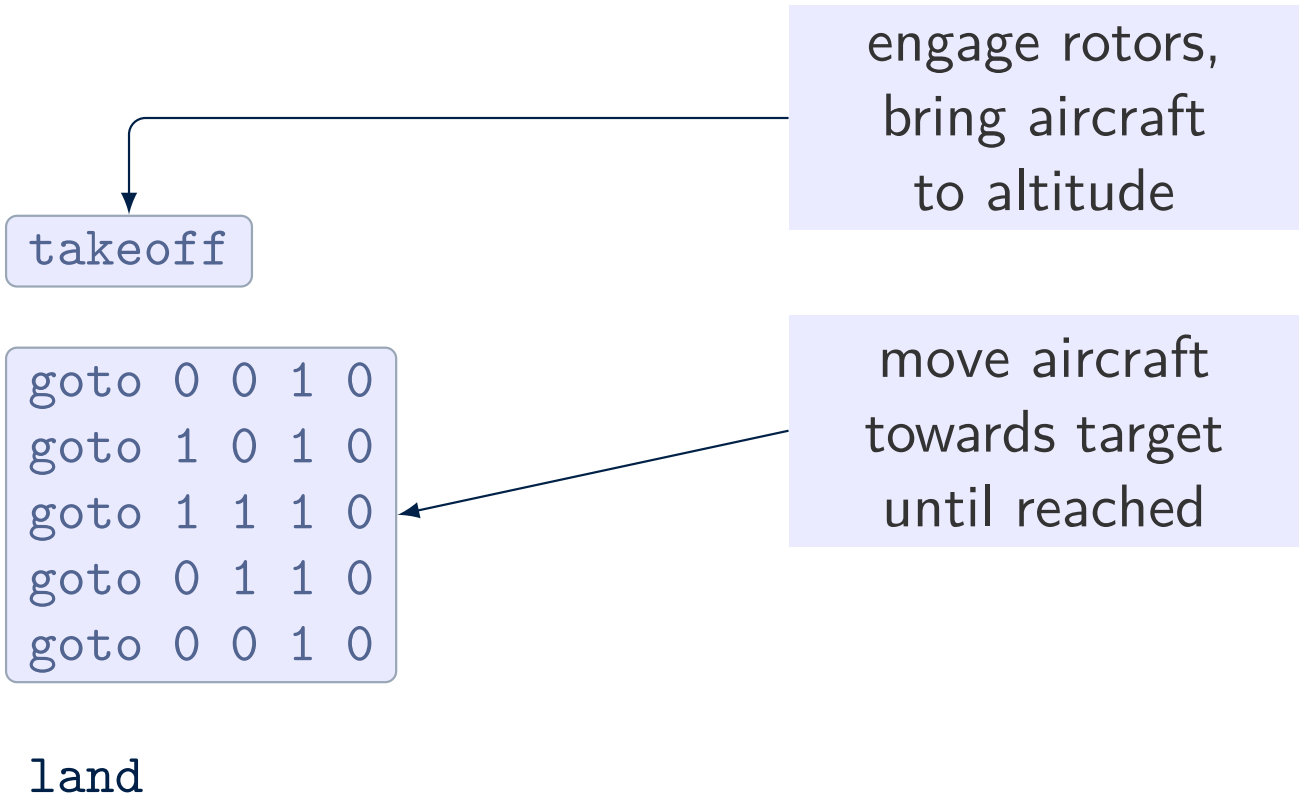

```
takeoff

goto 0 0 1 0
goto 1 0 1 0
goto 1 1 1 0
goto 0 1 1 0
goto 0 0 1 0

land
```

Sample Flight Plan

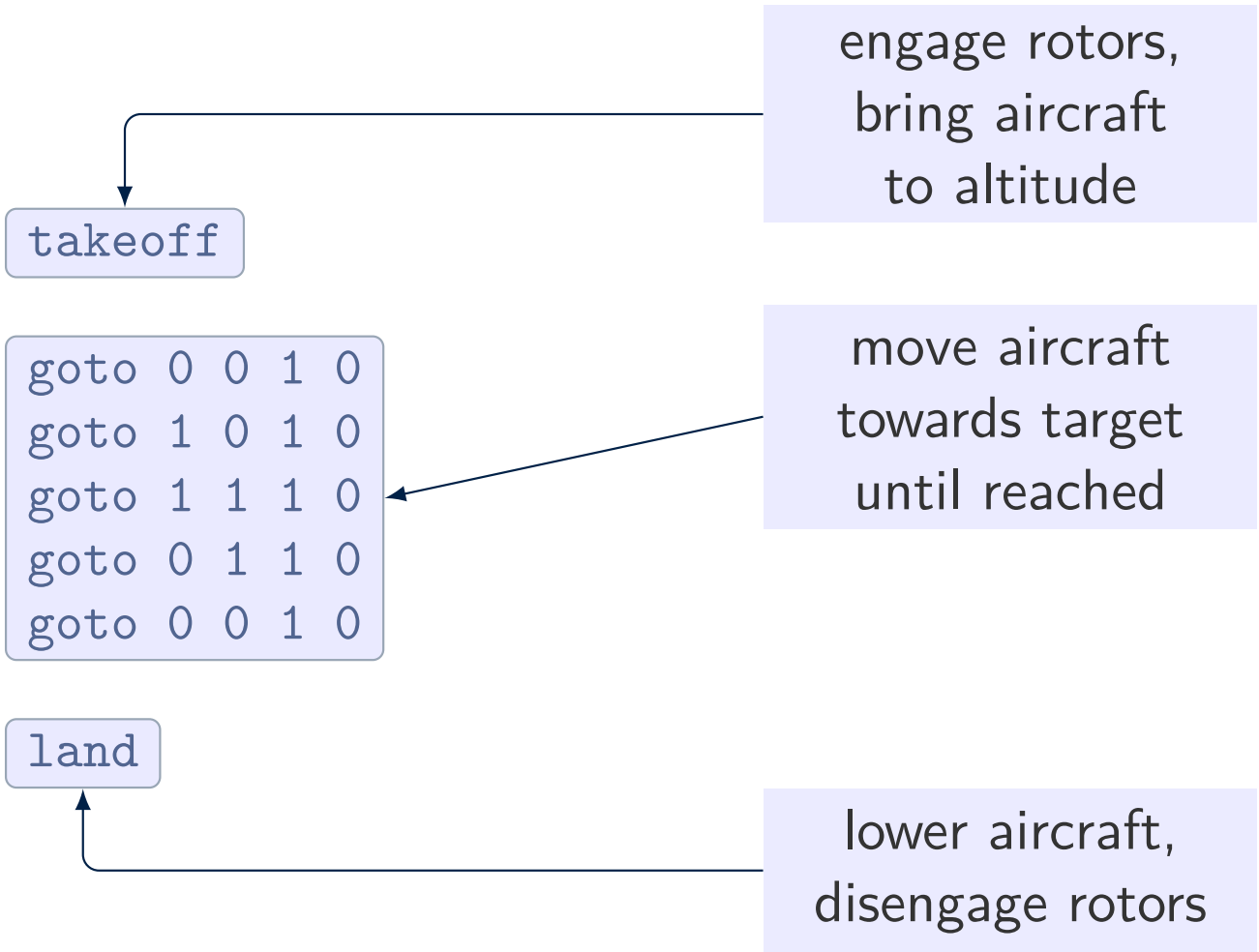
engage rotors, bring aircraft to altitude

```
takeoff

goto 0 0 1 0
goto 1 0 1 0
goto 1 1 1 0
goto 0 1 1 0
goto 0 0 1 0

land
```

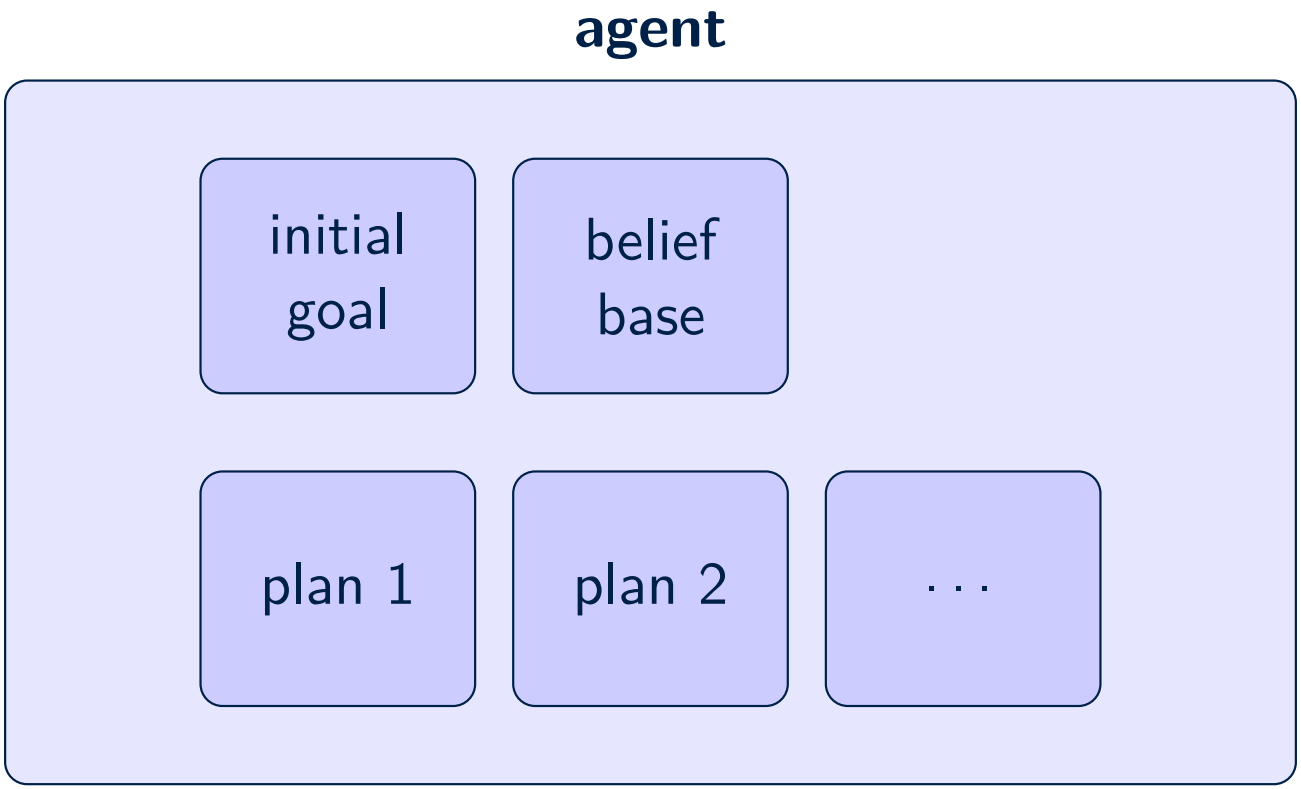
Sample Flight Plan



Sample Flight Plan



AgentSpeak



Plans


```
event
  :  condition & condition & ... & condition
  <- formula;
     formula;
     ...
     formula.
```

Plans

triggering events the plan is relevant for

```
event
  :  condition & condition & ... & condition
  <- formula;
     formula;
     ...
     formula.
```

Plans


```
event
  : condition & condition & ... & condition
  <- formula;
     formula;
     ...
     formula.
```

preconditions when the plan is applicable

Plans

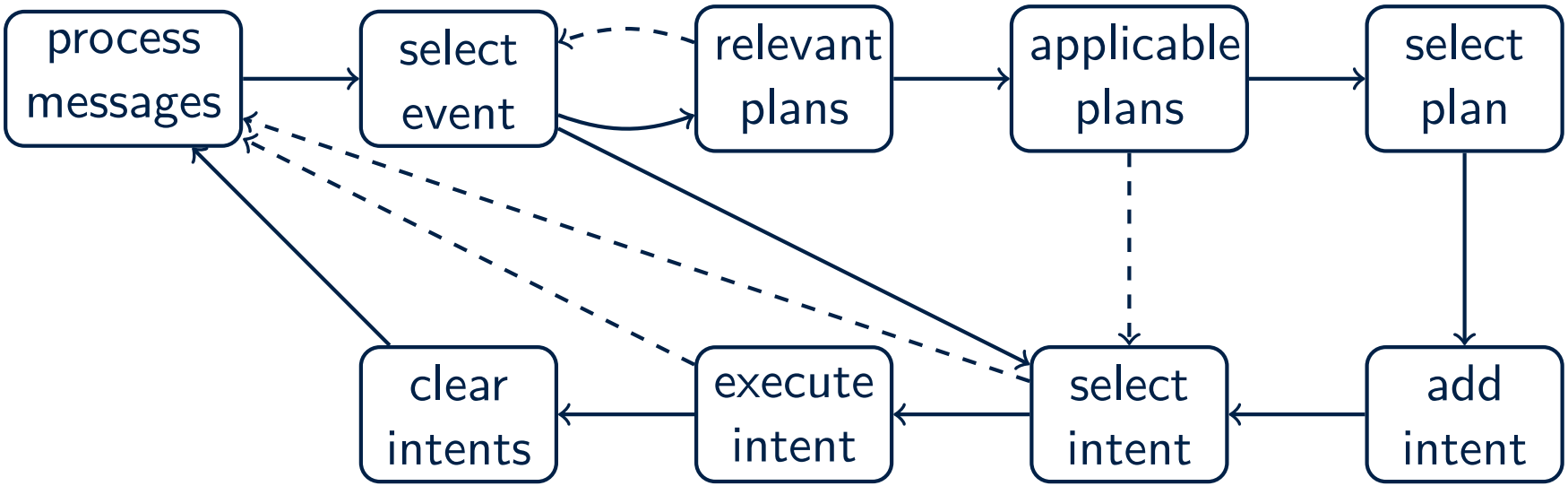

```
event
  :  condition & condition & ... & condition
  <- formula;
     formula;
     ...
     formula.
```

actual body of the plan to be executed

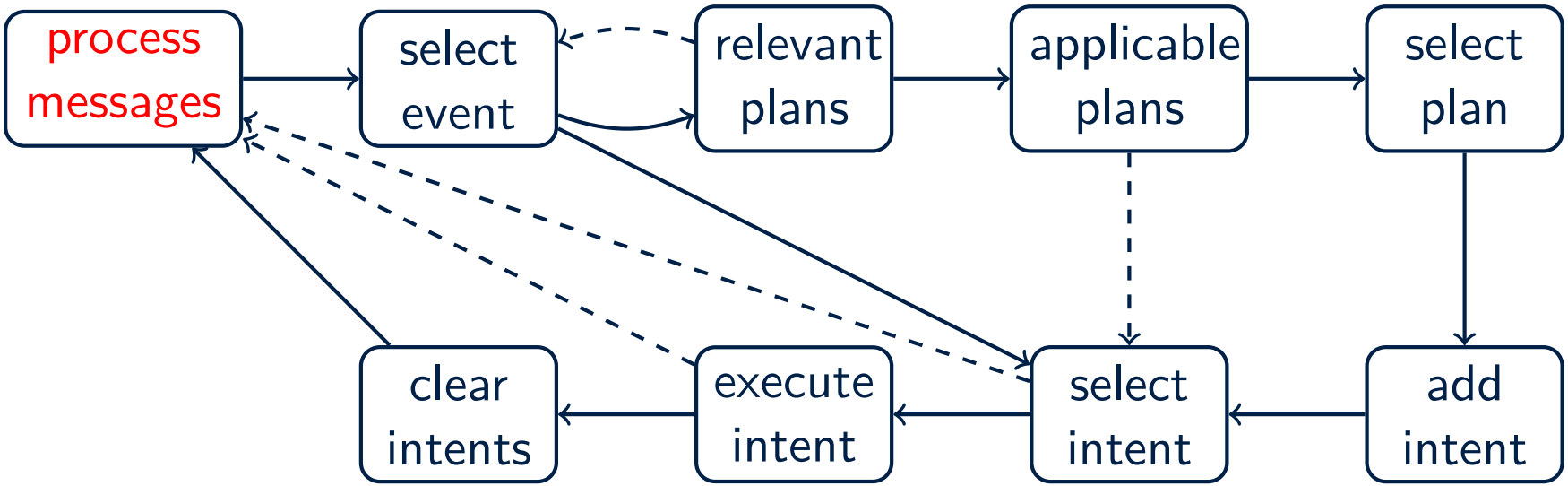
Plans, concretely


```
+!goto(Target) : takenOff
   <- sendHover();
      +lastTarget(Target);
      !completeGoto.

+!completeGoto
   : takenOff & myPosition(Pos)
     & lastTarget(Target) & not closeEnough(Pos, Target)
   <- Movement = calculateMovement(Pos, Target);
      sendControl(Movement);
      !completeGoto.

+!completeGoto
   : takenOff & myPosition(Pos)
     & lastTarget(Target) & closeEnough(Pos, Target)
   <- notifyUser("target reached").
```

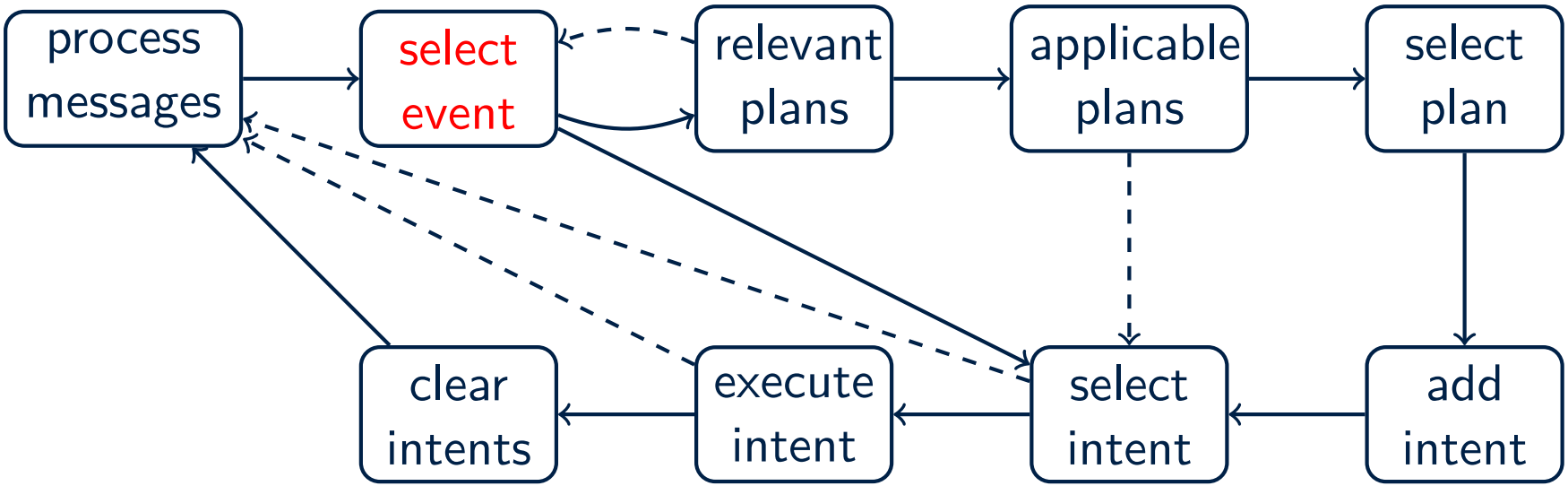
Semantics


process messages
select event
relevant plans
applicable plans
select plan
clear intents
execute intent
select intent
add intent

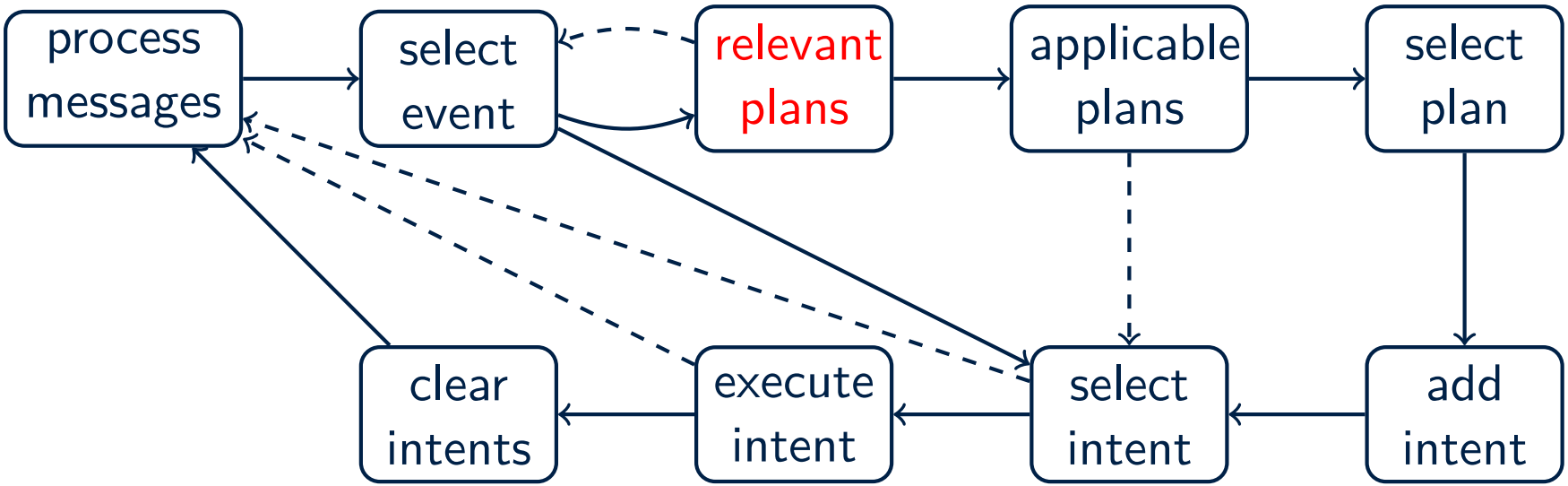
Semantics


process messages
select event
relevant plans
applicable plans
select plan
clear intents
execute intent
select intent
add intent

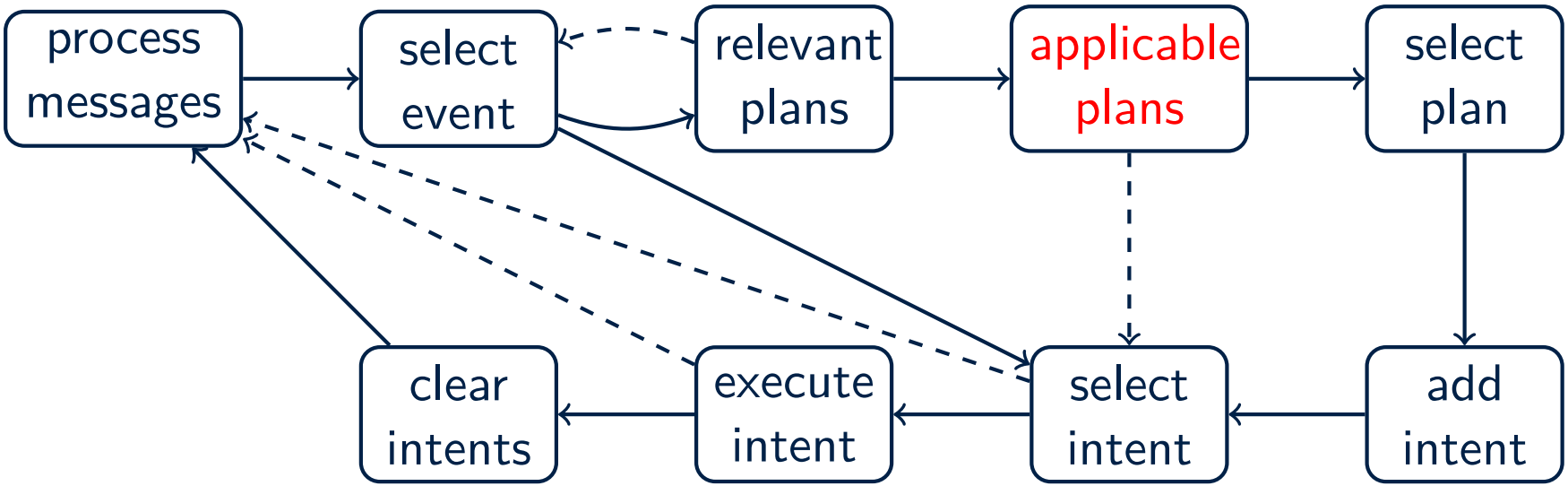
Semantics


process messages
select event
relevant plans
applicable plans
select plan
clear intents
execute intent
select intent
add intent

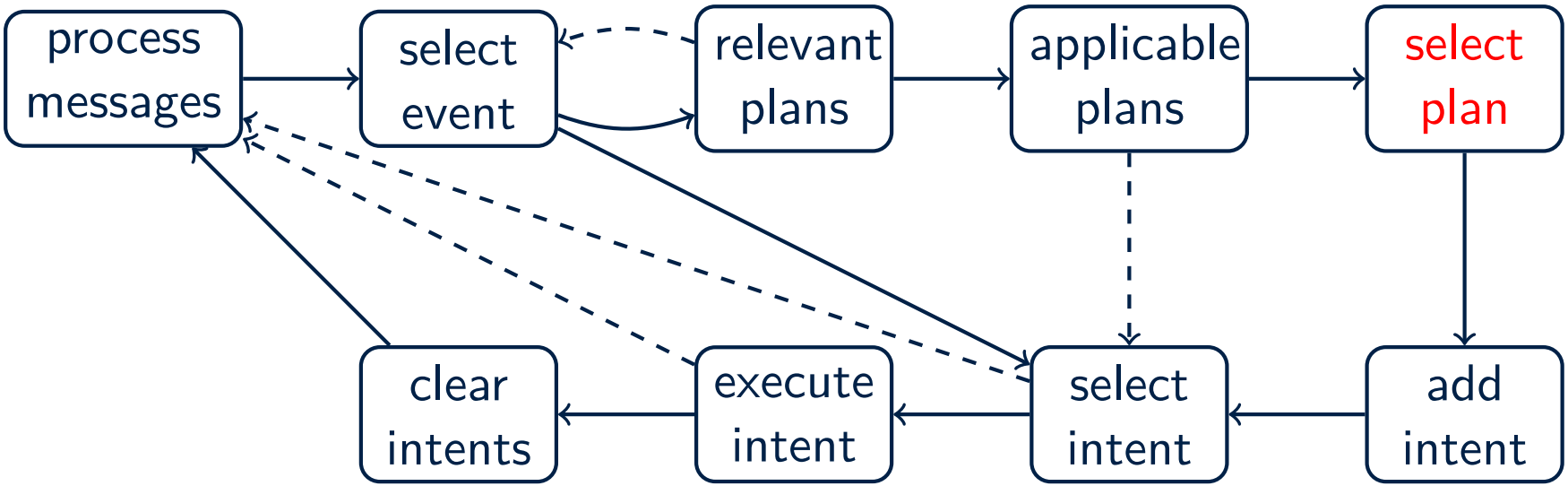
Semantics


process messages
select event
relevant plans
applicable plans
select plan
clear intents
execute intent
select intent
add intent

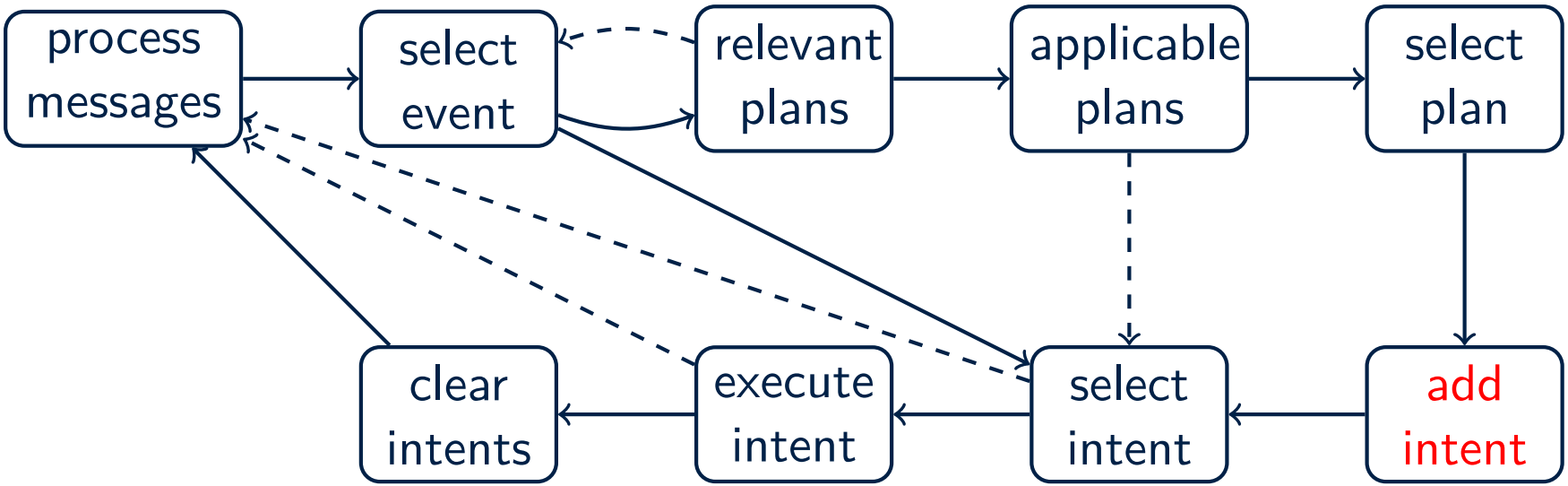
Semantics


process messages
select event
relevant plans
applicable plans
select plan
clear intents
execute intent
select intent
add intent

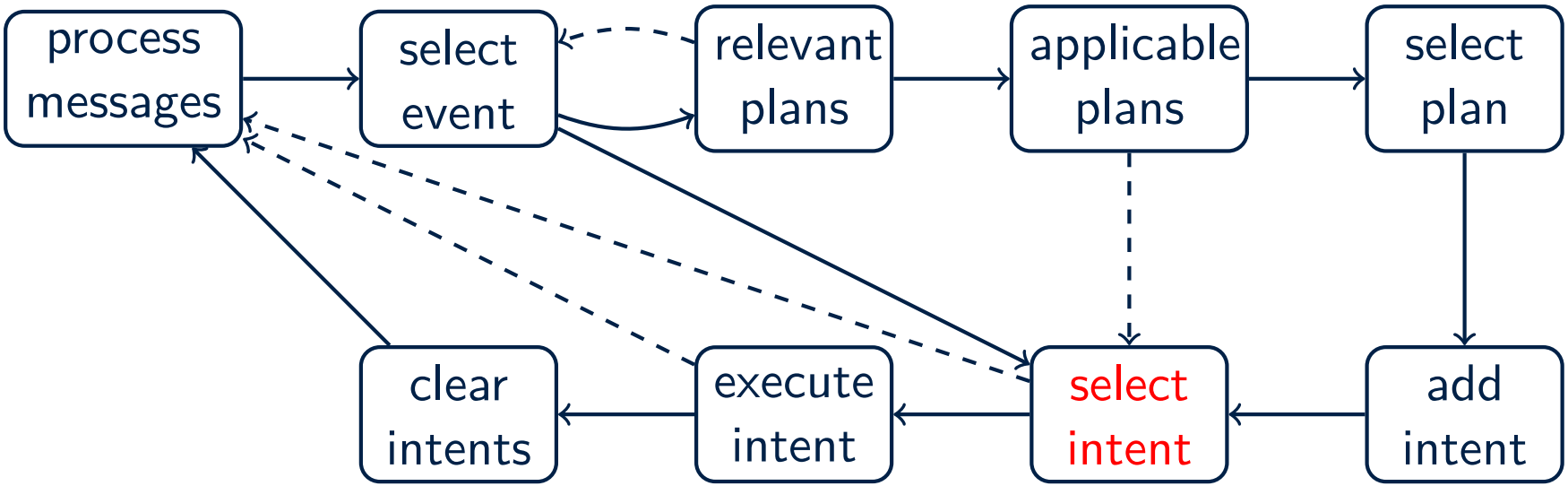
Semantics


process messages
select event
relevant plans
applicable plans
select plan
clear intents
execute intent
select intent
add intent

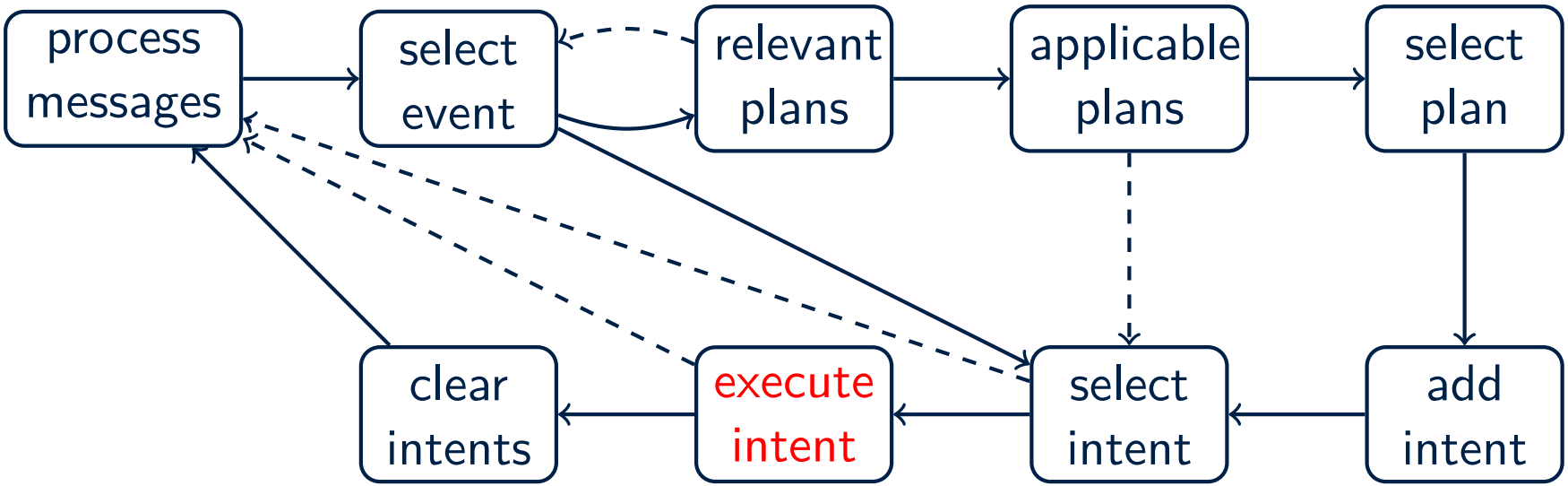
Semantics


process messages
select event
relevant plans
applicable plans
select plan
clear intents
execute intent
select intent
add intent

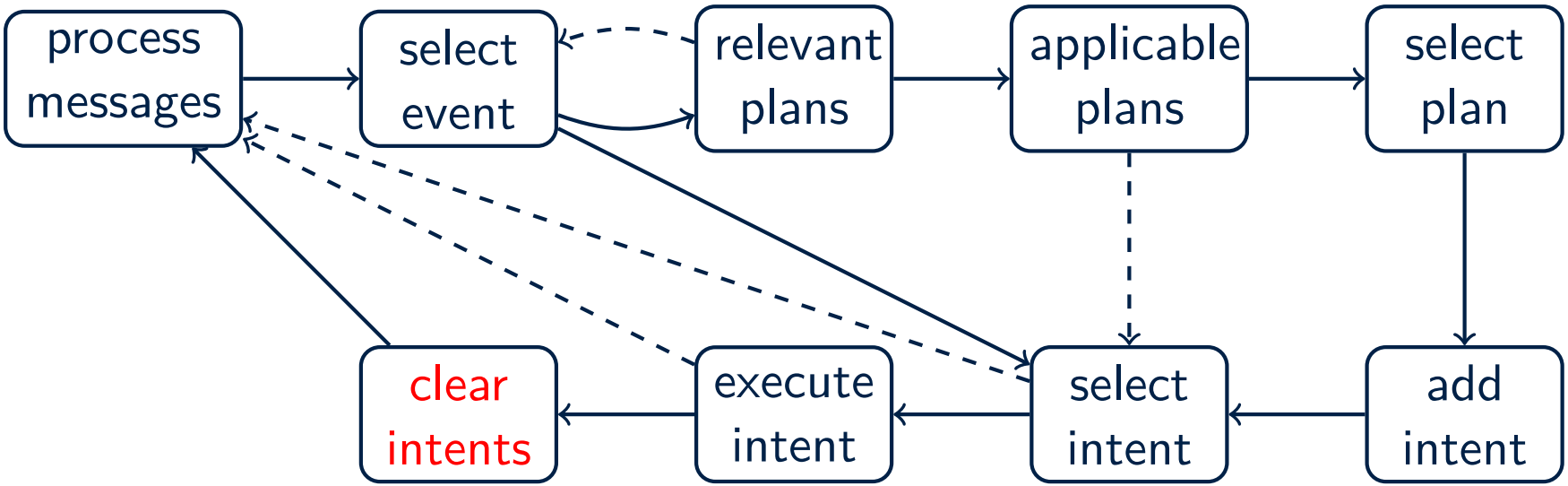
Semantics


process messages
select event
relevant plans
applicable plans
select plan
clear intents
execute intent
select intent
add intent

Semantics


process messages
select event
relevant plans
applicable plans
select plan
clear intents
execute intent
select intent
add intent

Semantics


process messages
select event
relevant plans
applicable plans
select plan
clear intents
execute intent
select intent
add intent

Our Customization

- "run-to-completion" style scheduling,
- only "tail recursion" allowed,
- belief base holds at most one instance of a literal

Translation

Relevant Plans Selection

```
void next_step(void) {
  updateBeliefs();
  eventt event = get_next_event();
  switch (event.trigger) {
    /* ... */
    case ADD_ACHIEVE_GOTO:
      add_achieve_goto(event.goto_param0);
      break;
    case ADD_ACHIEVE_COMPLETEGOTO:
      add_achieve_completeGoto();
      break;
    /* ... */
  }
}
```

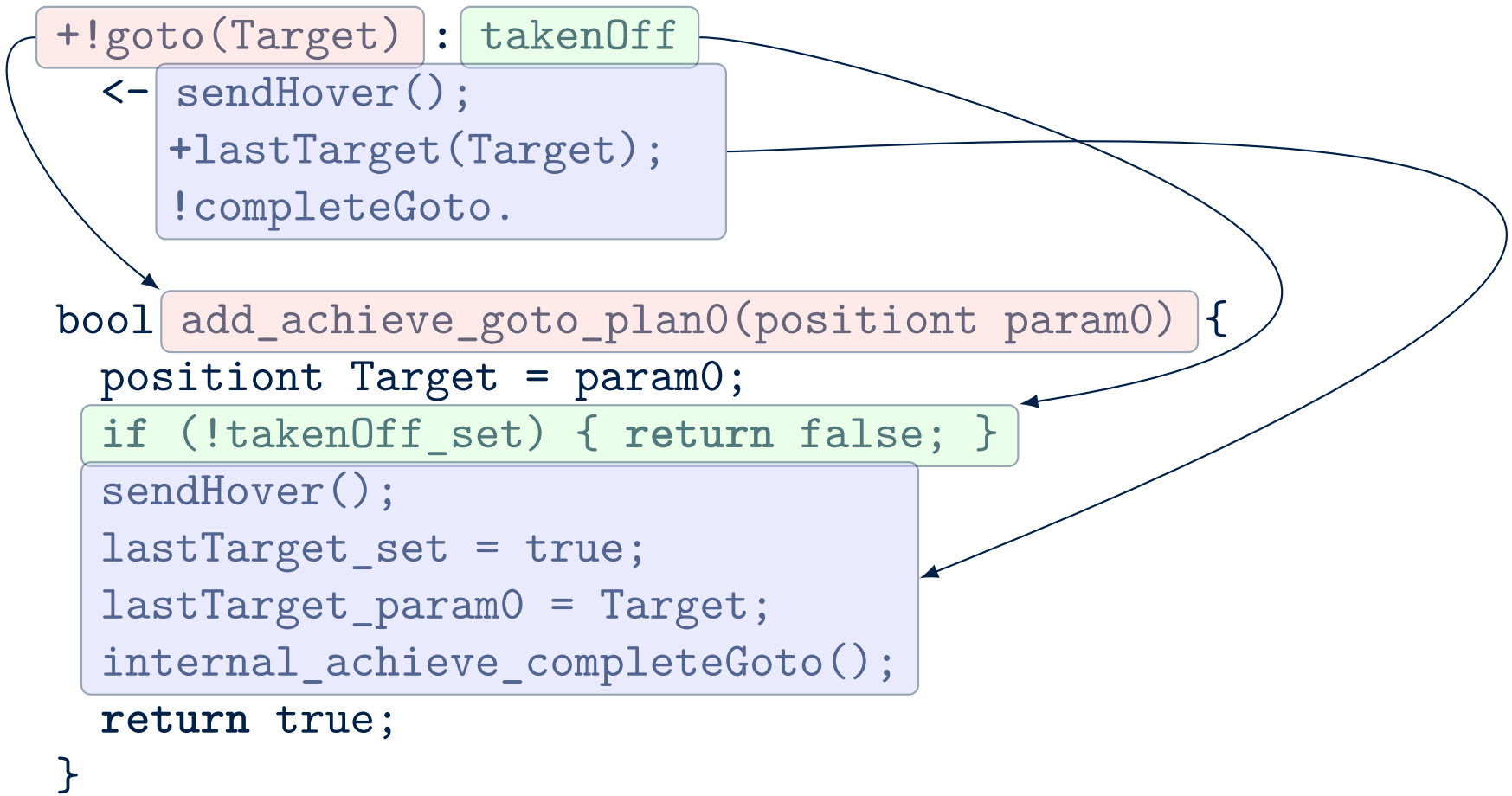
Translation

Applicable Plan Selection

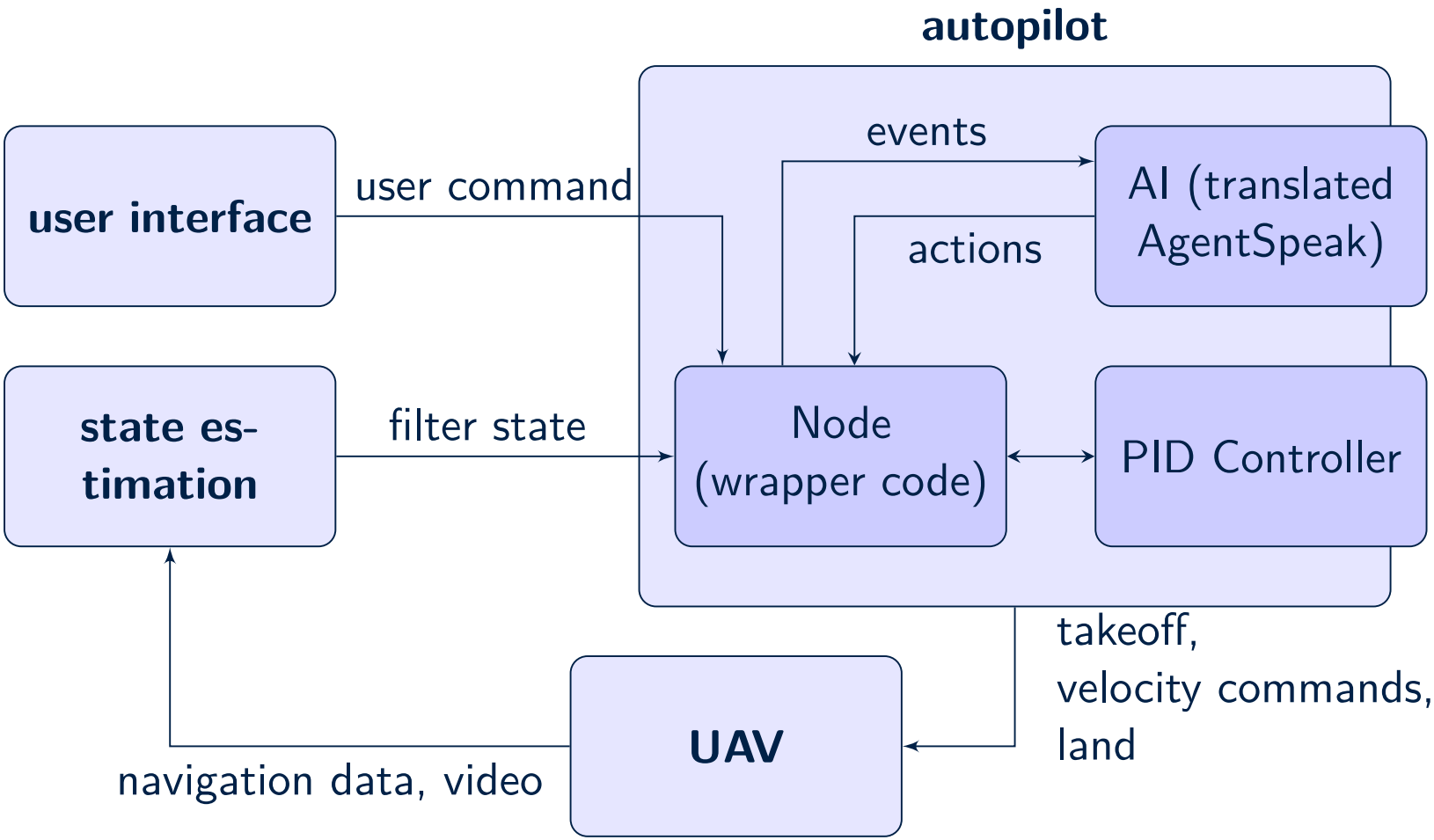
```
void add_achieve_goto(positiont param0) {
  /* try first plan */
  if (add_achieve_completeGoto_plan0()) {
    return;
  }
  /* try second plan */
  if (add_achieve_completeGoto_plan1()) {
    return;
  }

  /* ... handle the case where no plan is applicable ... */

  return;
}
```

Translation

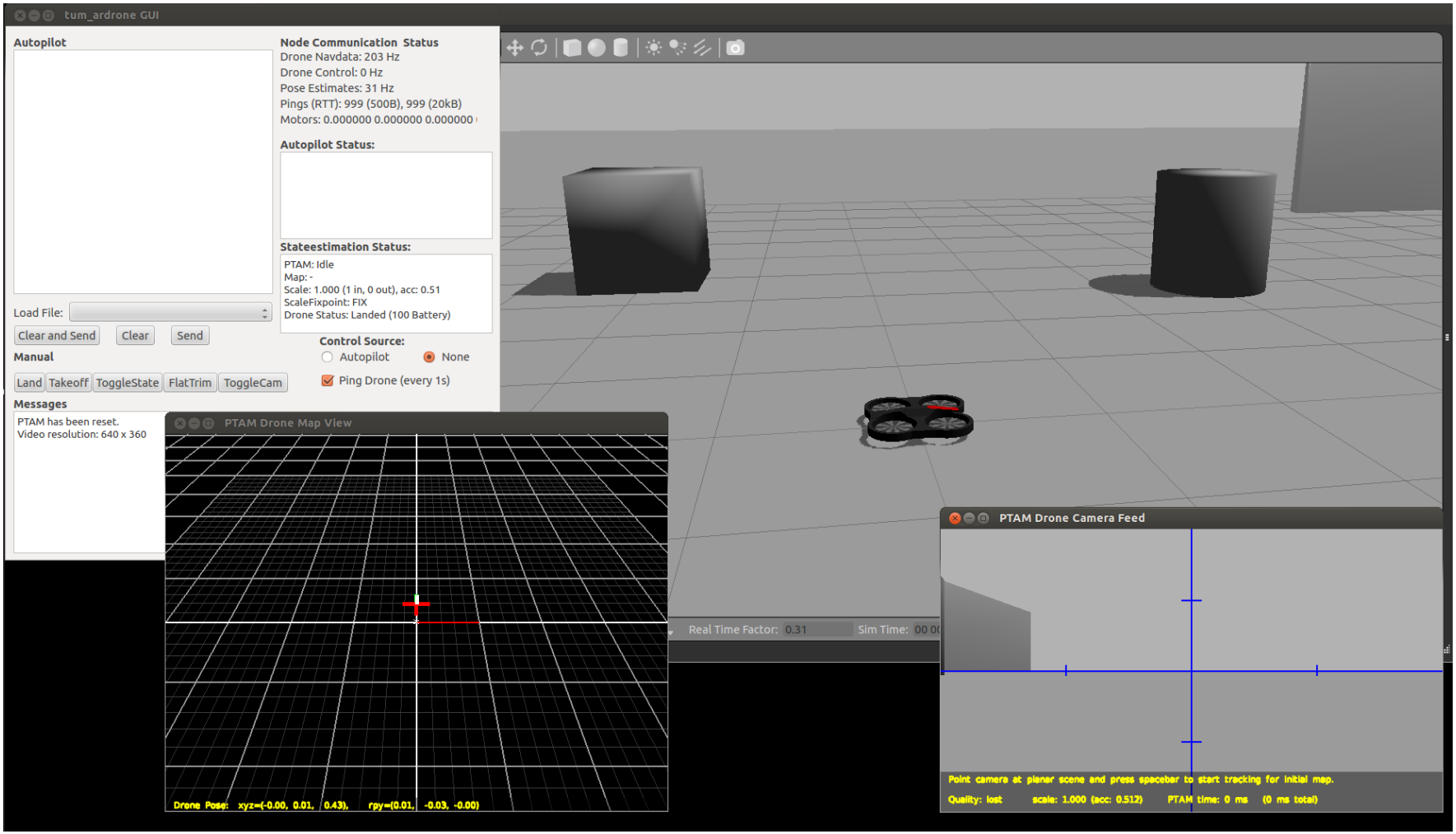
Plans



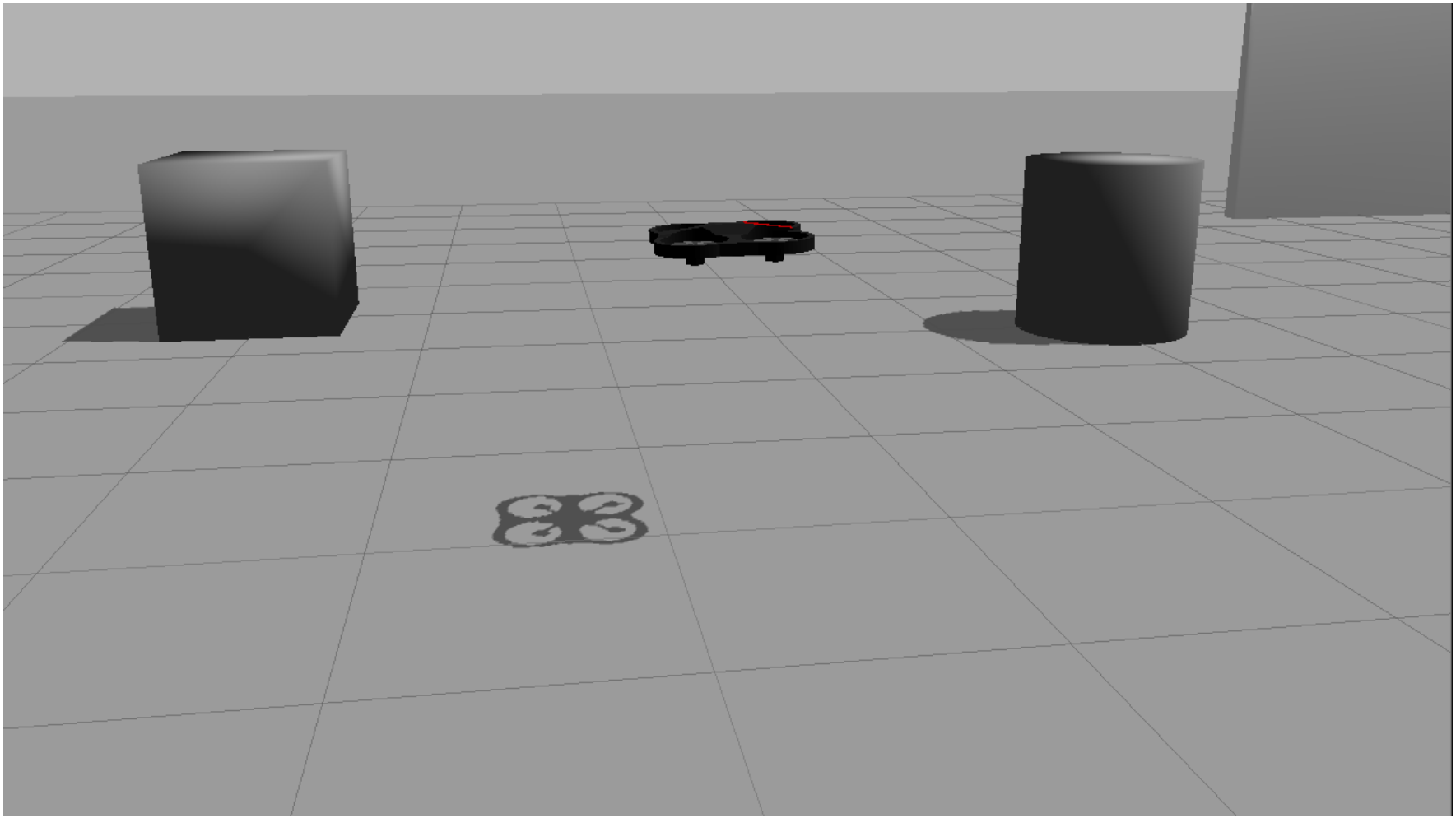
Experimental Setup



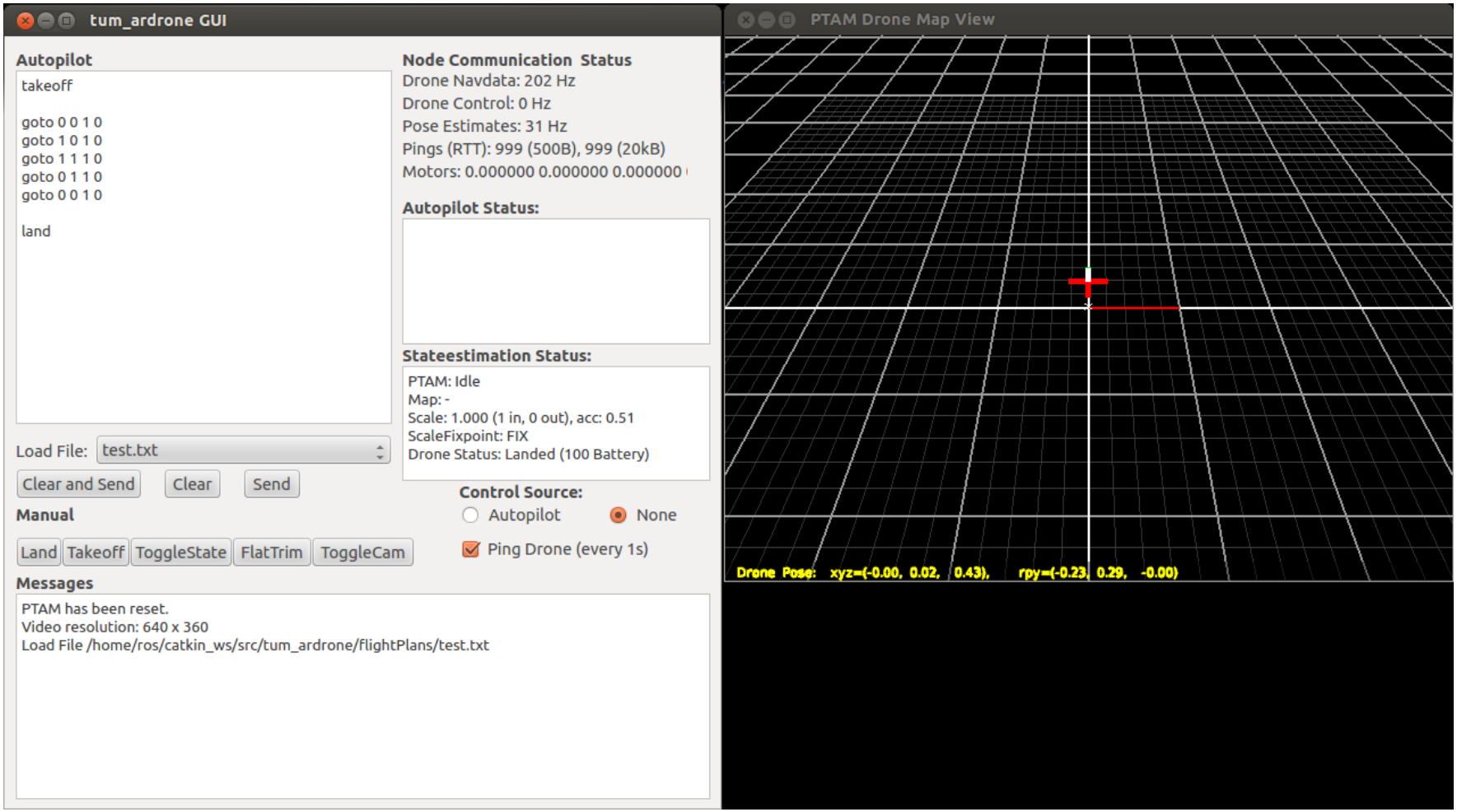
Test Flight


tum_ardrone GUI
Autopilot
Node Communication Status
Drone Navdata: 203 Hz
Drone Control: 0 Hz
Pose Estimates: 31 Hz
Pings (RTT): 999 (500B), 999 (20kB)
Motors: 0.000000 0.000000 0.000000
Autopilot Status:
Stateestimation Status:
PTAM: Idle
Map: -
Scale: 1.000 (1 in, 0 out), acc: 0.51
ScaleFixpoint: FIX
Drone Status: Landed (100 Battery)
Load File:
Clear and Send
Clear
Send
Control Source:
Autopilot
None
Manual
Land
Takeoff
ToggleState
FlatTrim
ToggleCam
Ping Drone (every 1s)
Messages
PTAM has been reset.
Video resolution: 640 x 360
PTAM Drone Map View
Drone Pose: xyz=(-0.00, 0.01, 0.43), rpy=(0.01, -0.03, -0.00)
Real Time Factor: 0.31
Sim Time:
PTAM Drone Camera Feed
Point camera at planer scene and press spacebar to start tracking for initial map.
Quality: lost
scale: 1.000 (acc: 0.512)
PTAM time: 0 ms
(0 ms total)

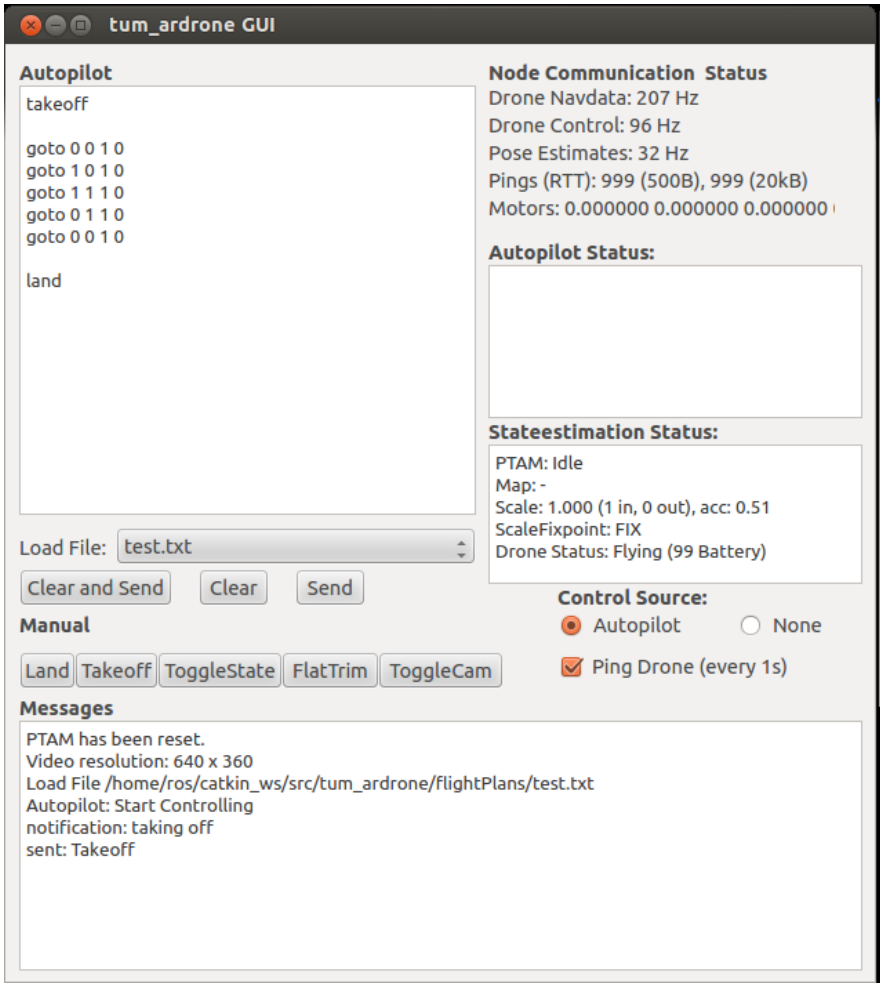
Test Flight

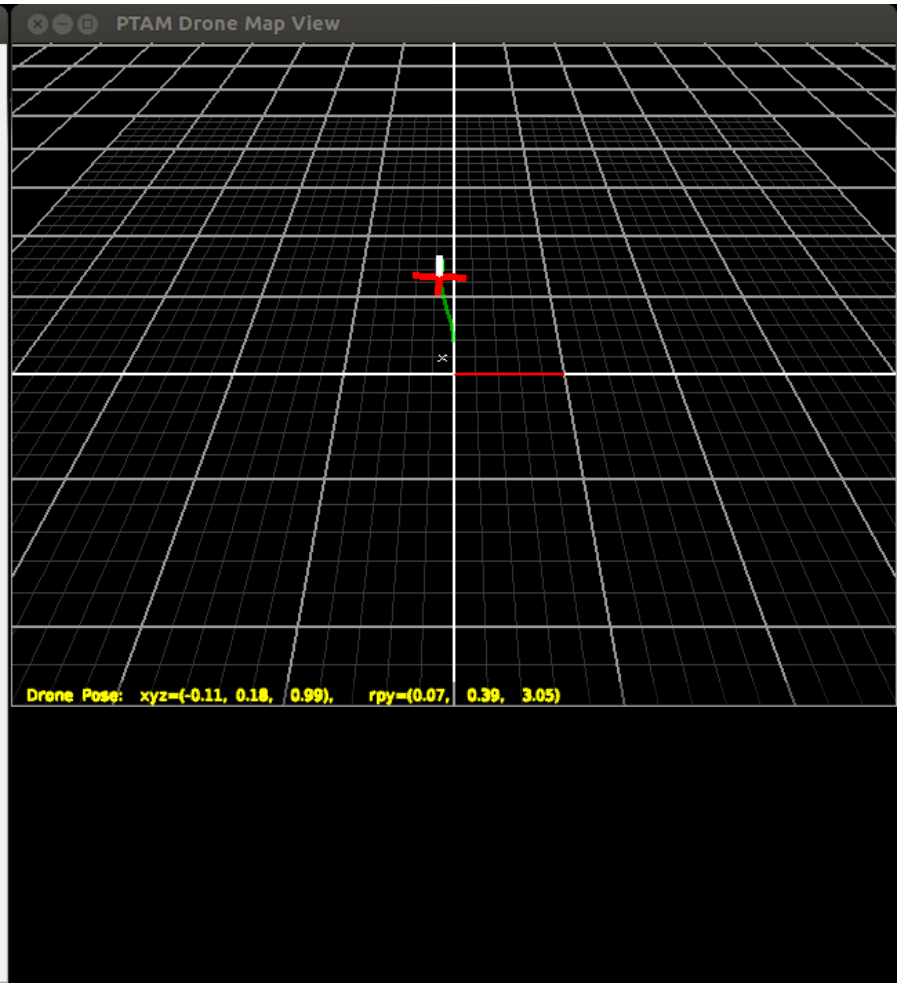


Test Flight

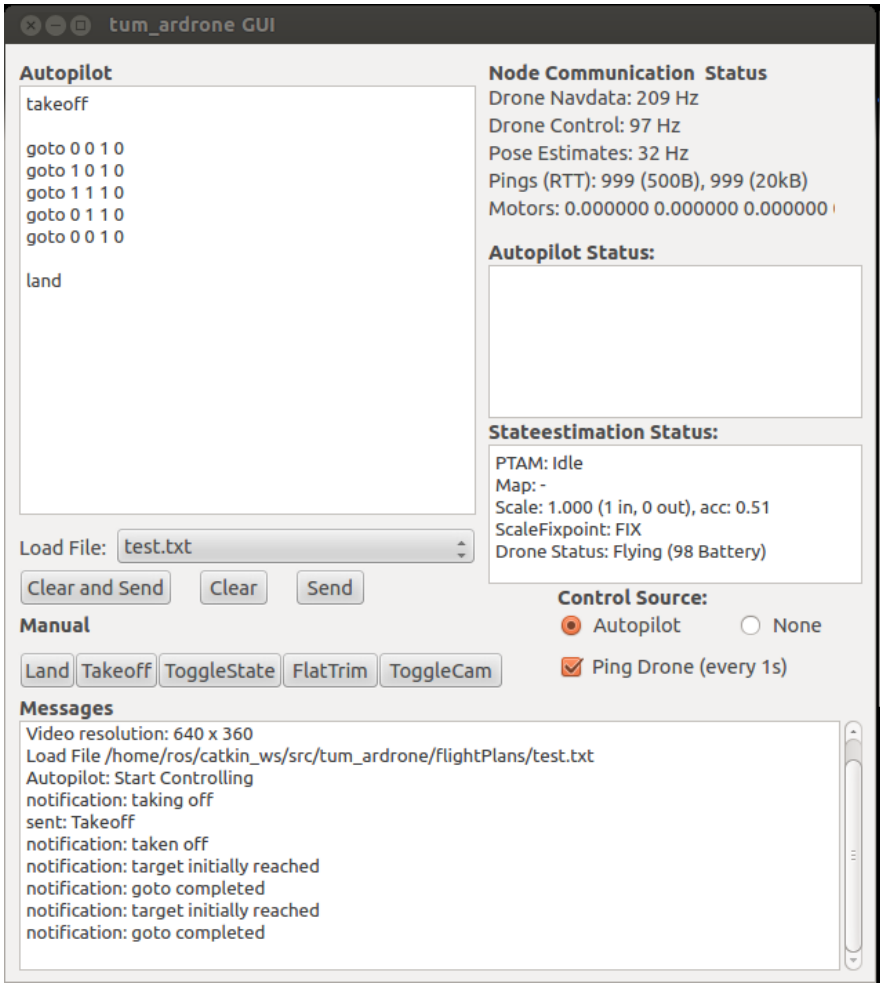

tum_ardrone GUI
Autopilot
takeoff
goto 0 0 1 0
goto 1 0 1 0
goto 1 1 1 0
goto 0 1 1 0
goto 0 0 1 0
land
Load File: test.txt
Clear and Send
Clear
Send
Manual
Land
Takeoff
ToggleState
FlatTrim
ToggleCam
Messages
PTAM has been reset.
Video resolution: 640 x 360
Load File /home/ros/catkin_ws/src/tum_ardrone/flightPlans/test.txt
Node Communication Status
Drone Navdata: 202 Hz
Drone Control: 0 Hz
Pose Estimates: 31 Hz
Pings (RTT): 999 (500B), 999 (20kB)
Motors: 0.000000 0.000000 0.000000
Autopilot Status:
Stateestimation Status:
PTAM: Idle
Map: -
Scale: 1.000 (1 in, 0 out), acc: 0.51
ScaleFixpoint: FIX
Drone Status: Landed (100 Battery)
Control Source:
Autopilot
None
Ping Drone (every 1s)
PTAM Drone Map View
Drone Pose: xyz=(-0.00, 0.02, 0.43), rpy=(-0.23, 0.29, -0.00)

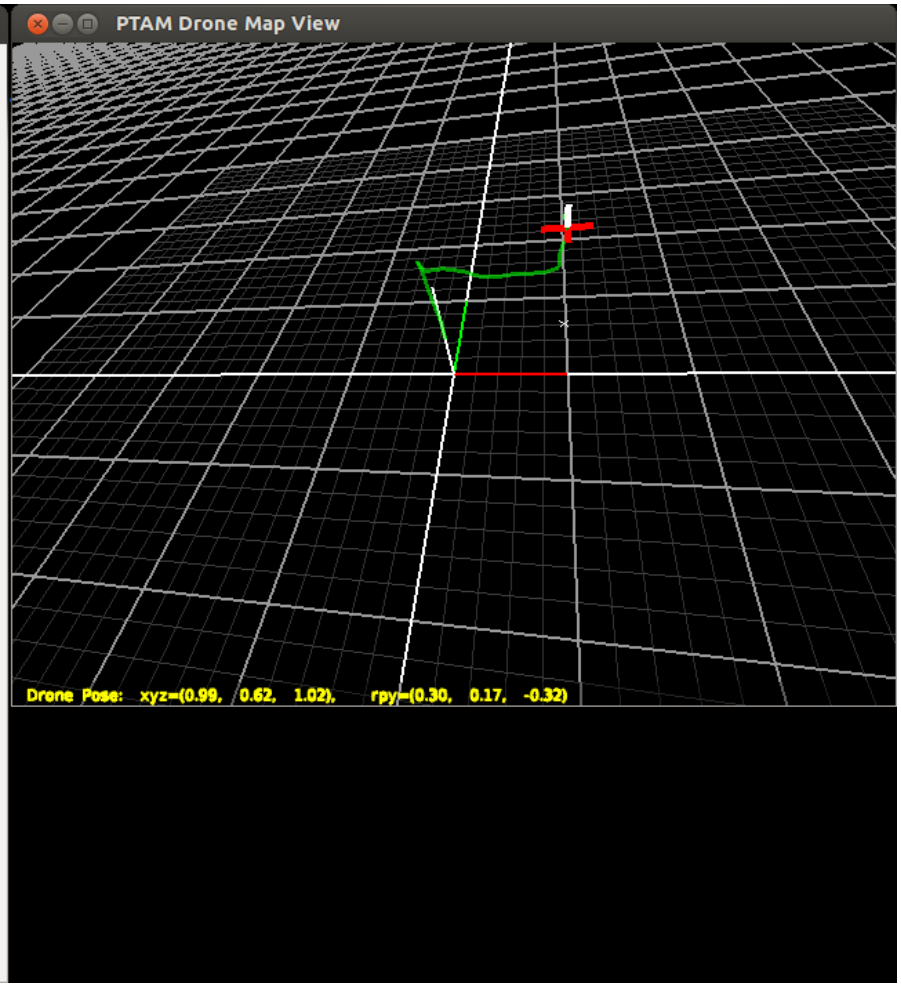
Test Flight


tum_ardrone GUI
Autopilot
takeoff
goto 0 0 1 0
goto 1 0 1 0
goto 1 1 1 0
goto 0 1 1 0
goto 0 0 1 0
land
Load File:
test.txt
Clear and Send
Clear
Send
Manual
Land
Takeoff
ToggleState
FlatTrim
ToggleCam
Node Communication Status
Drone Navdata: 207 Hz
Drone Control: 96 Hz
Pose Estimates: 32 Hz
Pings (RTT): 999 (500B), 999 (20kB)
Motors: 0.000000 0.000000 0.000000
Autopilot Status:
Stateestimation Status:
PTAM: Idle
Map: -
Scale: 1.000 (1 in, 0 out), acc: 0.51
ScaleFixpoint: FIX
Drone Status: Flying (99 Battery)
Control Source:
Autopilot
None
Ping Drone (every 1s)
Messages
PTAM has been reset.
Video resolution: 640 x 360
Load File /home/ros/catkin_ws/src/tum_ardrone/flightPlans/test.txt
Autopilot: Start Controlling
notification: taking off
sent: Takeoff

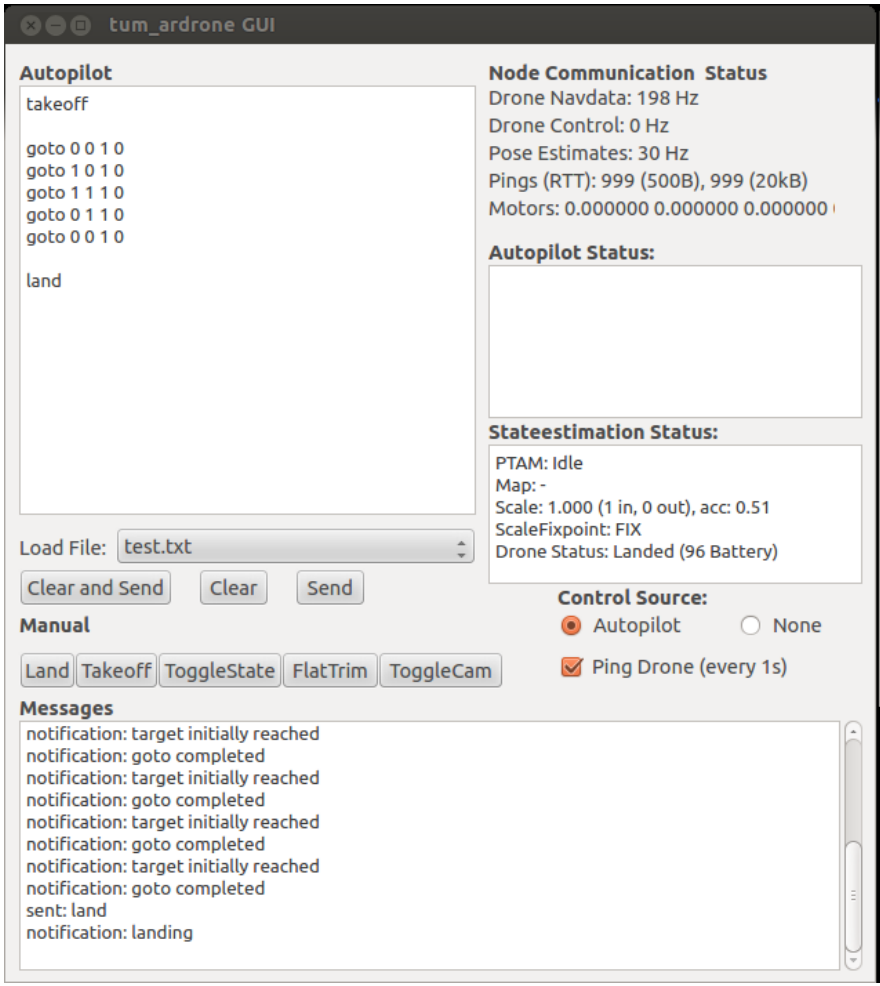

PTAM Drone Map View
Drone Pose: xyz=(-0.11, 0.18, 0.99), rpy=(0.07, 0.39, 3.05)

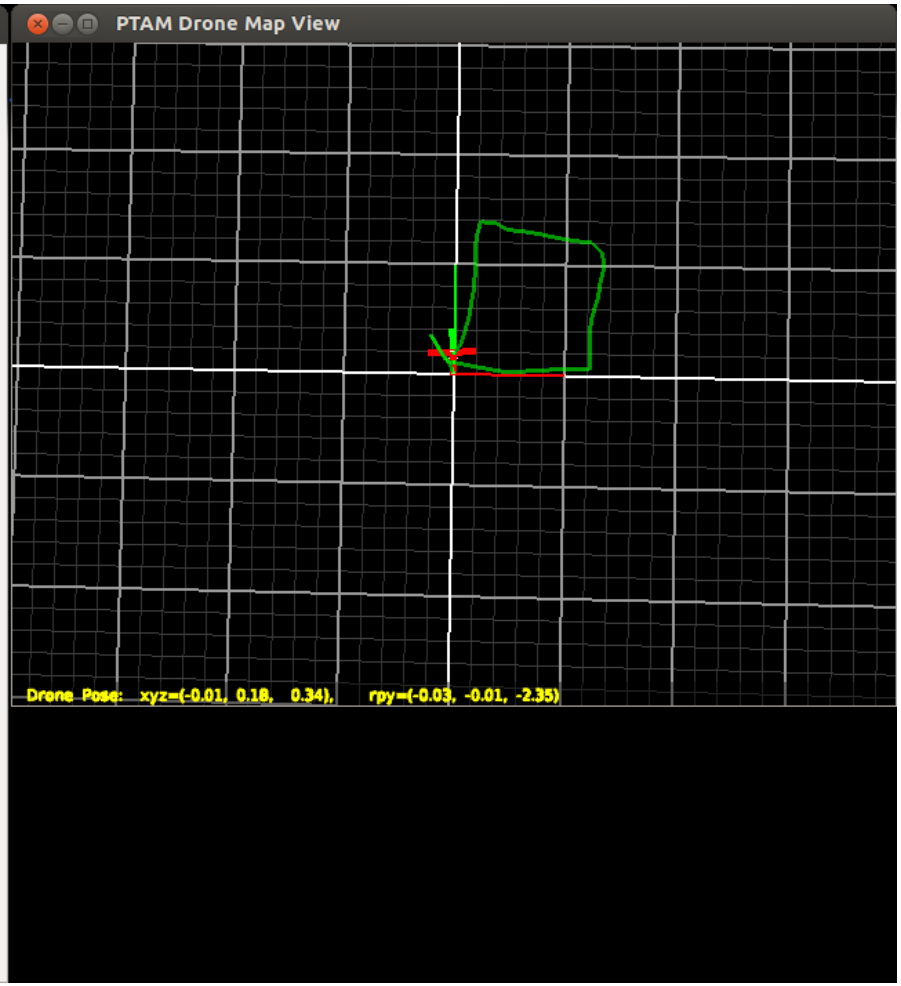
Test Flight


tum_ardrone GUI
Autopilot
takeoff
goto 0 0 1 0
goto 1 0 1 0
goto 1 1 1 0
goto 0 1 1 0
goto 0 0 1 0
land
Load File:
test.txt
Clear and Send
Clear
Send
Manual
Land
Takeoff
ToggleState
FlatTrim
ToggleCam
Node Communication Status
Drone Navdata: 209 Hz
Drone Control: 97 Hz
Pose Estimates: 32 Hz
Pings (RTT): 999 (500B), 999 (20kB)
Motors: 0.000000 0.000000 0.000000
Autopilot Status:
Stateestimation Status:
PTAM: Idle
Map: -
Scale: 1.000 (1 in, 0 out), acc: 0.51
ScaleFixpoint: FIX
Drone Status: Flying (98 Battery)
Control Source:
Autopilot
None
Ping Drone (every 1s)
Messages
Video resolution: 640 x 360
Load File /home/ros/catkin_ws/src/tum_ardrone/flightPlans/test.txt
Autopilot: Start Controlling
notification: taking off
sent: Takeoff
notification: taken off
notification: target initially reached
notification: goto completed
notification: target initially reached
notification: goto completed

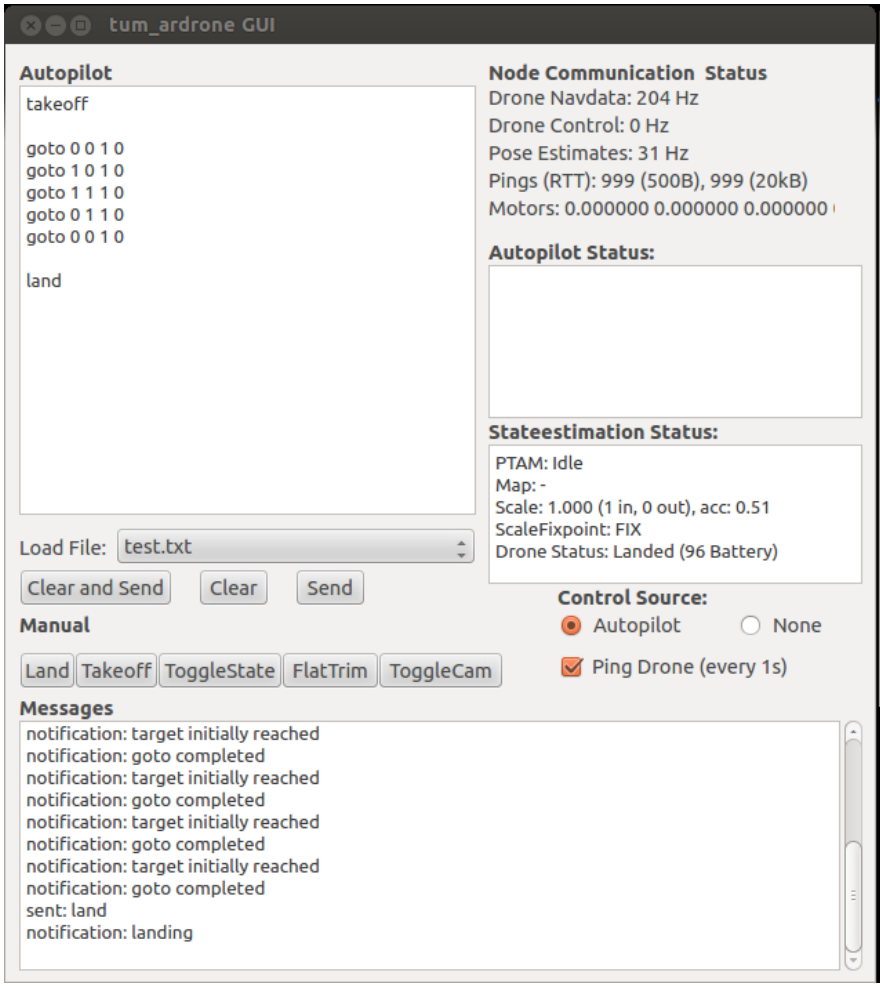

PTAM Drone Map View
Drone Pose: xyz=(0.99, 0.62, 1.02), rpy=(0.30, 0.17, -0.32)

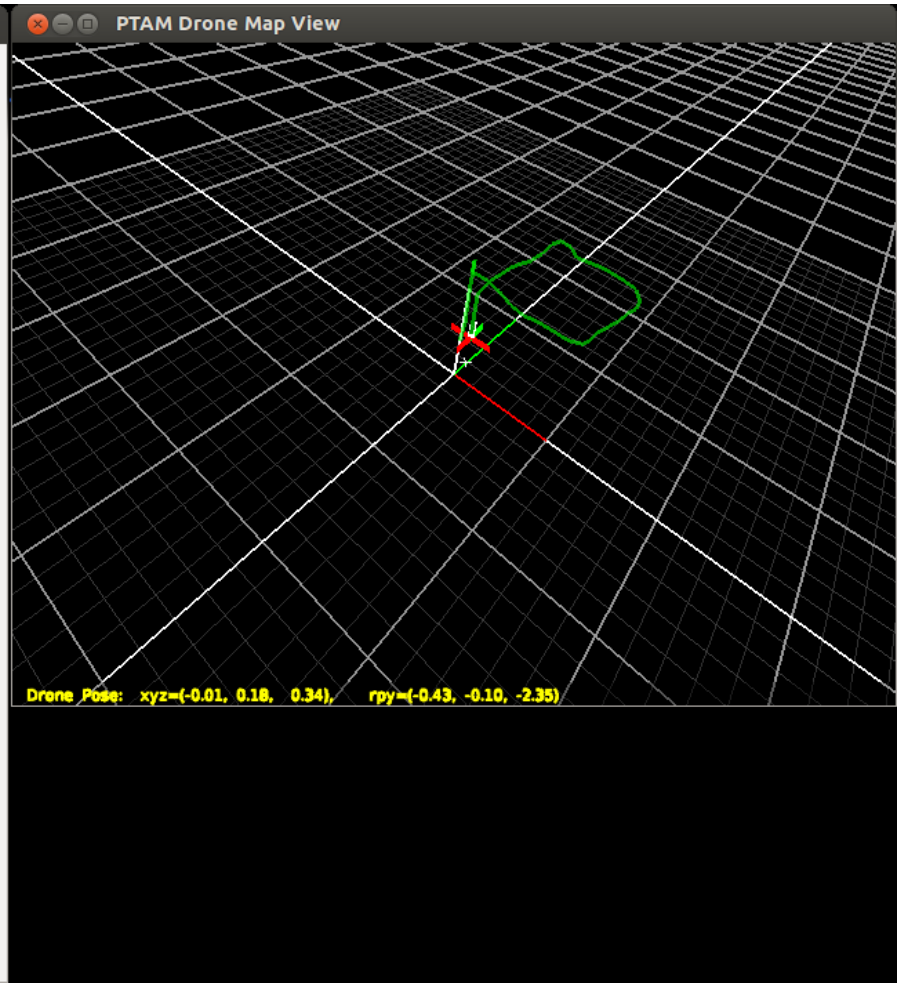
Test Flight


tum_ardrone GUI
Autopilot
takeoff
goto 0 0 1 0
goto 1 0 1 0
goto 1 1 1 0
goto 0 1 1 0
goto 0 0 1 0
land
Load File:
test.txt
Clear and Send
Clear
Send
Manual
Land
Takeoff
ToggleState
FlatTrim
ToggleCam
Node Communication Status
Drone Navdata: 198 Hz
Drone Control: 0 Hz
Pose Estimates: 30 Hz
Pings (RTT): 999 (500B), 999 (20kB)
Motors: 0.000000 0.000000 0.000000
Autopilot Status:
Stateestimation Status:
PTAM: Idle
Map: -
Scale: 1.000 (1 in, 0 out), acc: 0.51
ScaleFixpoint: FIX
Drone Status: Landed (96 Battery)
Control Source:
Autopilot
None
Ping Drone (every 1s)
Messages
notification: target initially reached
notification: goto completed
notification: target initially reached
notification: goto completed
notification: target initially reached
notification: goto completed
notification: target initially reached
notification: goto completed
sent: land
notification: landing


PTAM Drone Map View
Drone Pose: xyz=(-0.01, 0.18, 0.34), rpy=(-0.03, -0.01, -2.35)

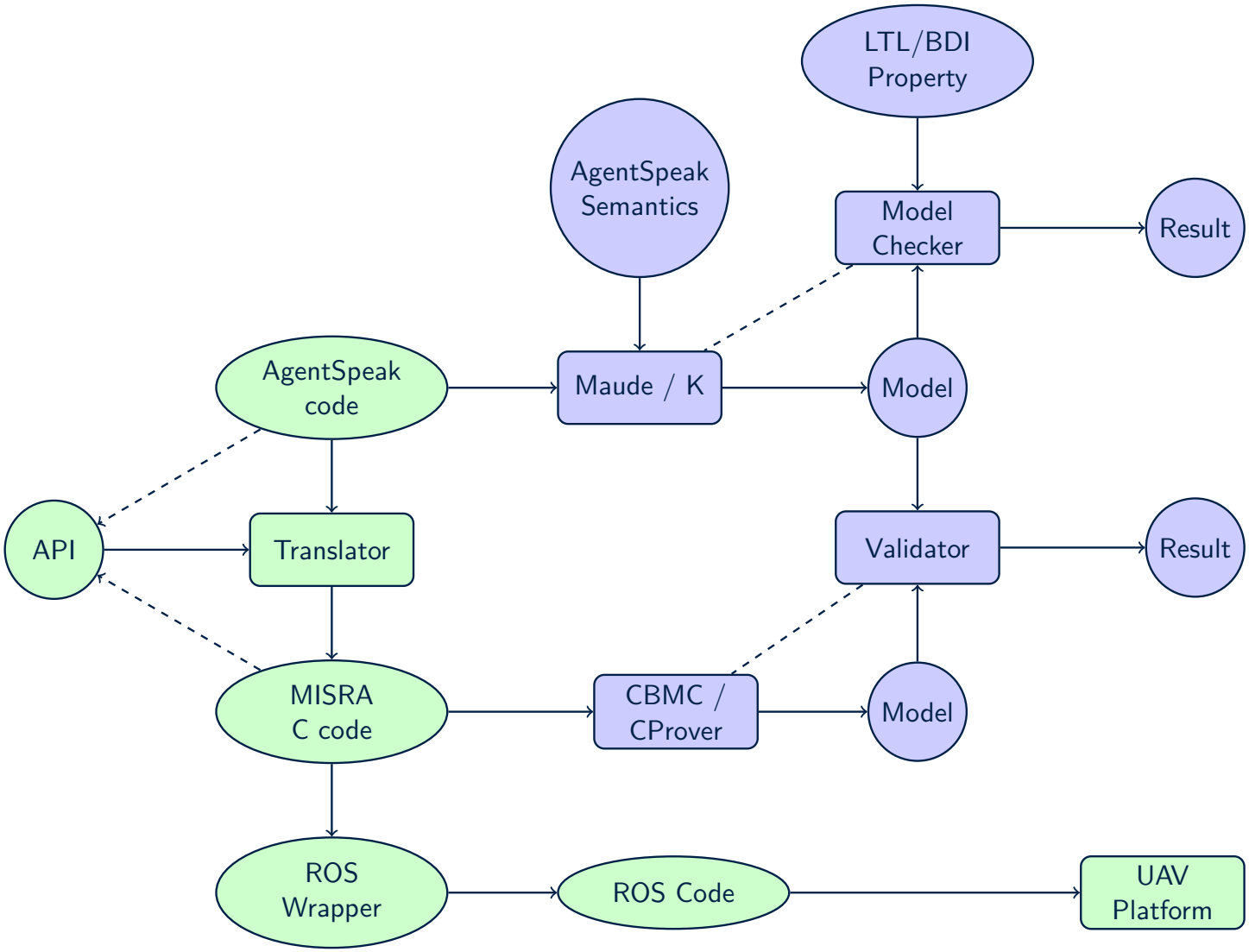
Test Flight


tum_ardrone GUI
Autopilot
takeoff
goto 0 0 1 0
goto 1 0 1 0
goto 1 1 1 0
goto 0 1 1 0
goto 0 0 1 0
land
Load File: test.txt
Clear and Send
Clear
Send
Manual
Land
Takeoff
ToggleState
FlatTrim
ToggleCam
Node Communication Status
Drone Navdata: 204 Hz
Drone Control: 0 Hz
Pose Estimates: 31 Hz
Pings (RTT): 999 (500B), 999 (20kB)
Motors: 0.000000 0.000000 0.000000
Autopilot Status:
Stateestimation Status:
PTAM: Idle
Map: -
Scale: 1.000 (1 in, 0 out), acc: 0.51
ScaleFixpoint: FIX
Drone Status: Landed (96 Battery)
Control Source:
Autopilot
None
Ping Drone (every 1s)
Messages
notification: target initially reached
notification: goto completed
notification: target initially reached
notification: goto completed
notification: target initially reached
notification: goto completed
notification: target initially reached
notification: goto completed
sent: land
notification: landing


PTAM Drone Map View
Drone Pose: xyz=(-0.01, 0.18, 0.34), rpy=(-0.43, -0.10, -2.35)

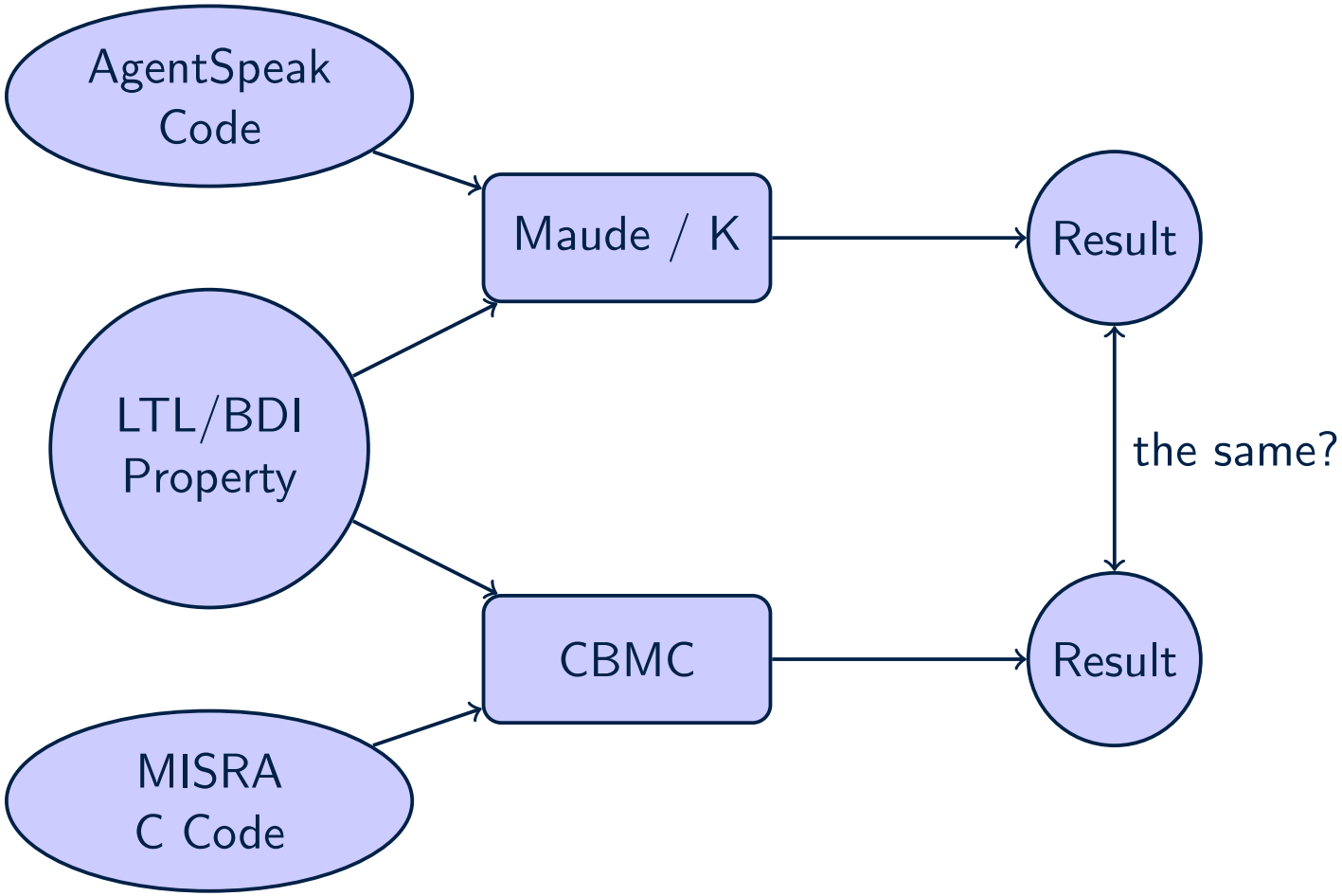
Test Flight

Future Work


LTL/BDI
Property
AgentSpeak
Semantics
Model
Checker
Result
AgentSpeak
code
Maude / K
Model
API
Translator
Validator
Result
MISRA
C code
CBMC /
CProver
Model
ROS
Wrapper
ROS Code
UAV
Platform

A Simple Translation Validation

UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE



Conclusions

- translation from agent-oriented programming language to low-level language + translation validation = traceability
- what is the sweet spot for agent-oriented programming languages: expressivity vs. translatability?
- translation validation?