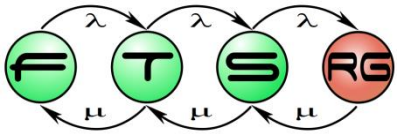
Cooperative Drones
Third Homework Assignment
DNS-2016

György Demarcsek

Krisztián Nagy

Ferenc Attila Somogyi

Budapesti Műszaki és Gazdaságtudományi Egyetem
Hibatűrő Rendszerek Kutatócsoport



Sirius ⇔ Xtext ⇔ Simulation

- Sirius
  - Edit any structural instance model
- Xtext
  - Import structural instance model
  - Generate behaviour model, not Java files
- Simulation
  - Behaviour is processed, events are „generated” into event chains
  - Event chaining instead of Java file generation

Simulation

- We simulate the execution of a drone's program (behav. model XMI files taken as input)
- "Executed" drone program instructions generate DESMO-J events
- The following features / properties are all taken into account:
  - Drone min. / max. velocity, CPU performance, memory size
  - Distance between drones (communication)
  - Drone battery drains on every instruction and recharges while nearby its charge station
  - Collision: drones, field objects
  - Drone action support; range of actions

Simulation Events

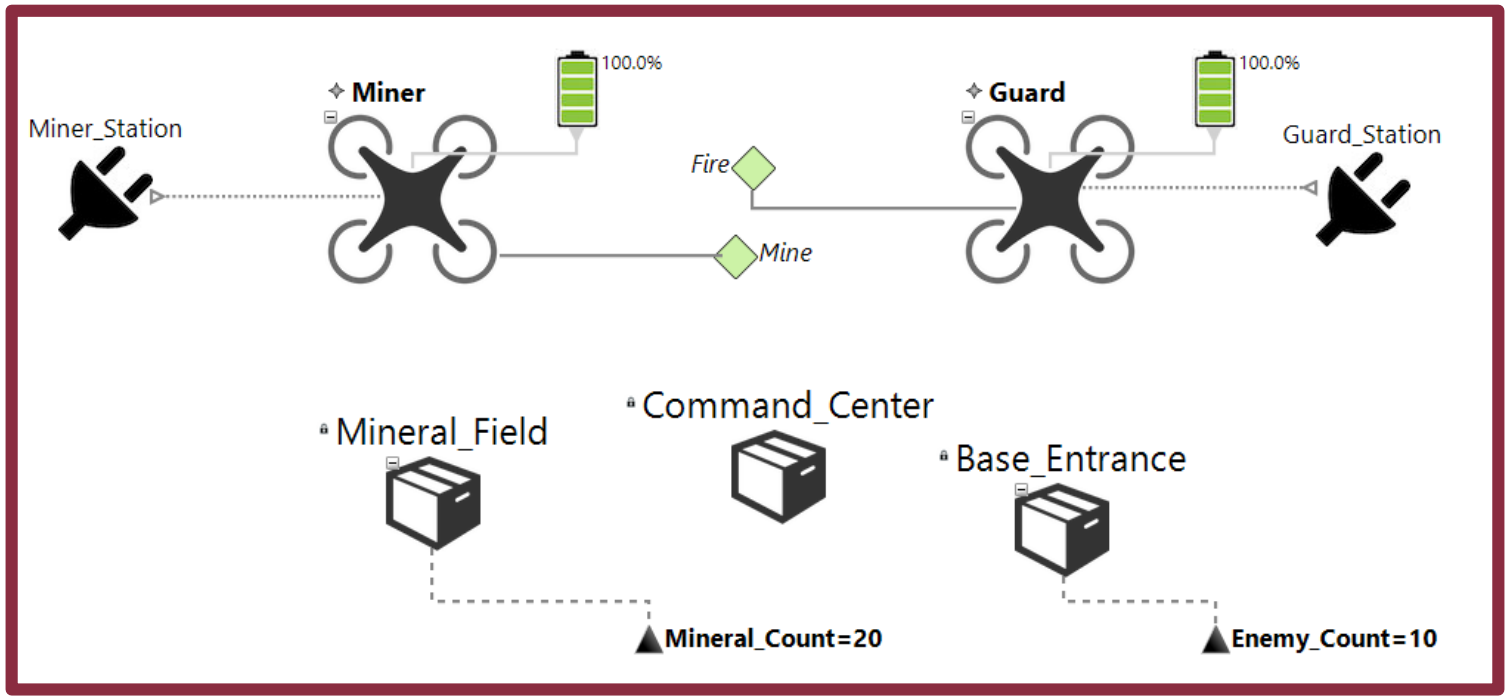
- Instruction Events
  - Move*, Choice, Lift, SendMessage, PerformAction, Instruct, …
- External Events
  - Communication
    - MessageDropped, MessageDelivered …
  - Timely
    - BatteryCharged, BatteryDepleted …
  - Accidental
    - DroneCollided, exceptions* …

Simulation Metrics

- DESMO-J Counters
  - Battery death
  - Message drop
  - Executed instructions
  - Number of collisions

Example – StarCraft

- Two drones
  - Miner
  - Guard
- Mineral field
  - Mineable
- Base Entrance
  - Enemies
- Charge stations
  - Full battery at start



Example – StarCraft

- Miner drone behaviour
  - While there are minerals
    - Move to mineral field
    - Mine
    - Return minerals
  - Return to charge station

```
import "platform:/resource/hu.bme.mit.mdsd.dns2016.drones.behaviour.instancemodels/starcraft.drones"

behav MinerBehav (interrupt = true);

drones { Miner }

// While there are still minerals left to mine
while (Mineral_Field.Mineral_Count > 0) {
    // Move to the mineral field, mine, then go back to the command center
    moveto Mineral_Field;
    action Mine on Mineral_Field;
    moveto Command_Center;
}

moveto Miner_Station;
```

Example – StarCraft

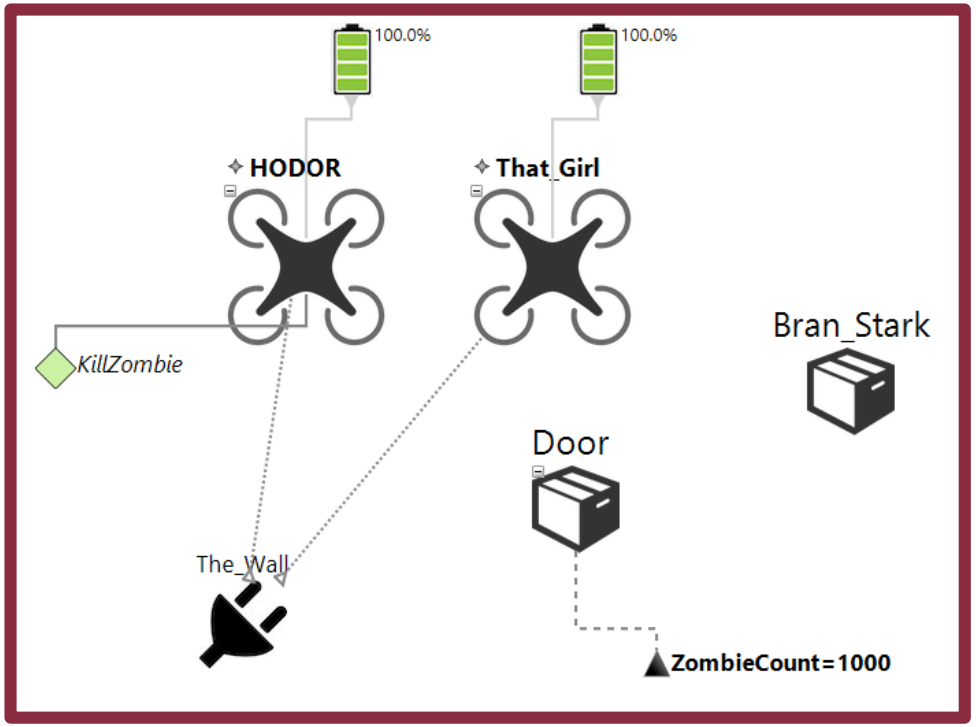
- Guard drone behaviour
  - While there are minerals
  - Move to mineral field
  - Mine
  - Return to base

```
behav GuardBehav (interrupt = false);
drones {Guard}

moveto Base_Entrance;

// While there are still minerals left to mine
while (Mineral_Field.Mineral_Count > 0) {

    // Move to base entrance and check if there are enemies nearby
    moveto Base_Entrance;
    if (Base_Entrance.Enemy_Count > 0) {

        // Instruct the miners to move to safety
        send instruct {
            moveto Command_Center;
            waitfor SafeToComeOut 20.0 {
                wait 0.5;
            } timeout {
                wait 0.5;
            }
        }

        // Fire at enemies
        while (Base_Entrance.Enemy_Count > 0) {
            action Fire on Base_Entrance;
            wait 1.0;
        }

        // Move back to the station
        moveto Guard_Station;
        wait 10.0;
    }
}

moveto Guard_Station;
```

Spoiler Alert!

Example – GoT

- Two drones
  - Hodor
  - That Girl
- The door
  - Must be held at all costs



Example – GoT

- That girl
  - Instruct Hodor
    - Hold the door!
    - Action KillZombie
  - Wait for response
    - Pick up Bran
    - Move to the Wall
- Hodor
  - Wait for message
    - Send response

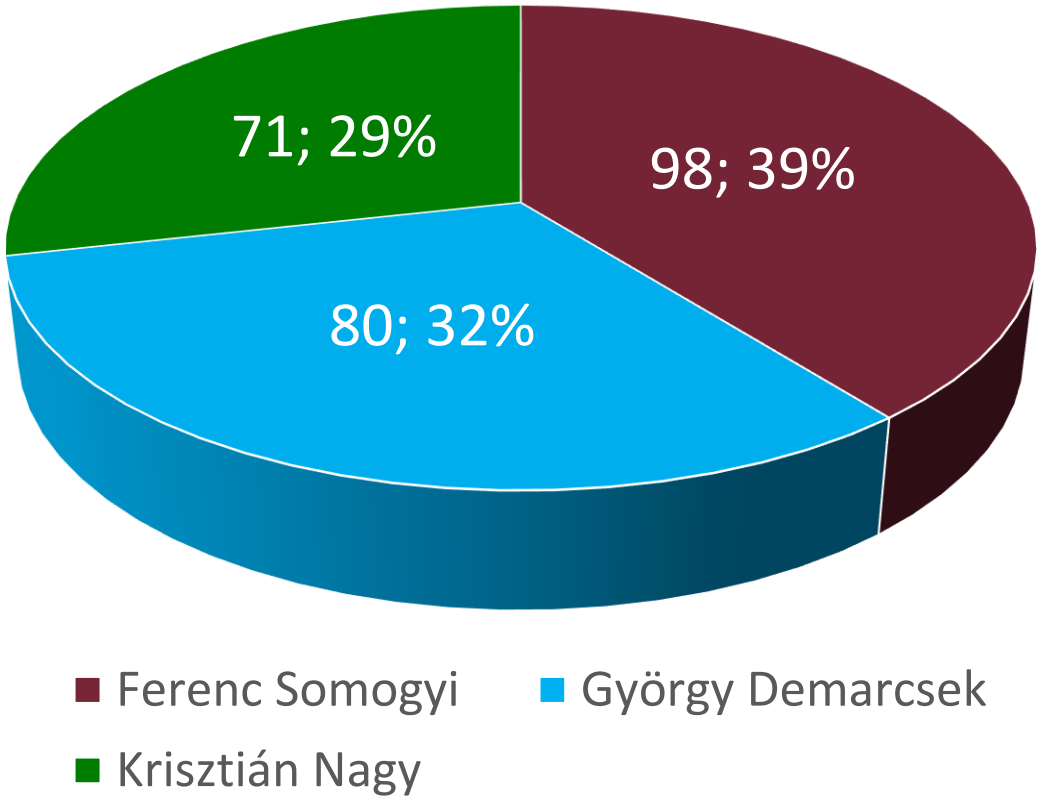
```
behav ThatGirl (interrupt = false);
drones { That_Girl }

wait 3.0;

send msg Hold_the_door;

waitfor HODOR 30.0 {
    lift Bran_Stark;
} timeout {
    send msg MEH;
}

send instruct {
    moveto Door;

    while (Door.ZombieCount > 0) {
        action KillZombie on Door;
    }

    send msg HODOR;
}

moveto The_Wall;
place object;
```

```
behav HodorBehav (interrupt = true);
drones { HODOR }

waitfor Hold_the_door 30.0 {
    send msg HODOR;
} timeout {
    wait 1.0;
}
```

Extras

- Definitive extras
  - Conditions and loops in the behav. model
  - Drone communication range + message queues
  - Moving obstacles (drone payload)
  - Dynamically modifable event chains (support for manual instructions and events)
- Has some value
  - Intuitive and easy-to-use behaviour programming with Xtext (+ validation)
  - Refined collision detection (and 3D movement)

Division of Work



Q & A