Hands-On Session One: Simulating Plan Execution

Outline

- Run the provided examples
- Write and run a simple plan with simulation script.

Running the PLEXIL examples

- Inspect the examples
  - Go to `plexil/examples`
  - Look at the plans and scripts in an editor (Emacs, etc)
    - `DriveToTarget, SafeDrive, SimpleDrive`
    - Same as above with `-script` suffix
- Compile the examples
  - `plexil <plan-file>`
  - `plexilscript <script-file>`
- Run the examples
  - Start LUV viewer: type `luv`
  - `run-ue <plan-file>`
    - Script file found automatically for examples.

Outline

- Run the provided examples
- Write and run a simple plan with simulation script.

Simulation Scripts

- A *simulation script* encodes *events* from the external world used in simulating plan execution.

```
initial-state {
  state At ("Rock" : string) = false : bool;
}
script {
  state At ("Rock" : string) = true : bool;
  command-success drive (1.0 : real);
  command-success takeSample ();
}
```

- What does this script do?
  - changes the `At` state to true (i.e. the rover has reached the rock).
  - acknowledges the `drive` command with the `COMMAND_SUCCESS` handle.
  - acknowledges the `takeSample` command with the `COMMAND_SUCCESS` handle.

Simulation Scripts (continued)

- Important point: when commands return values, the handle must occur *after* the value.

```
script {
  command         get-input () = "yes" : string;
  command-success get-input ();
}
```

  - Note convenient form `command-success`, which is used frequently.

Exercise One

- In this exercise you will:
  - Write a PLEXIL plan for the RoboSim application
  - Write a simulation script
  - Run the plan with simulation script, using LUV

Robot simulator application

- The application is called RoboSim
  - Move a robot in a two-dimensional space with obstacles, other moving robots, energy sources, and a goal.
  - The RoboSim world will be represented by a simulation script.
  - Actual RoboSim will be used in next exercise session.
- Interface is entirely commands and function calls (no lookups)
  - See the RoboSim handout

Write a PLEXIL plan and script

- The Plan
  - Determine current robot position.
  - Move the robot 4 steps
    - One step in each direction (up, down, right, left)
    - Any order
    - Assume move is successful
  - The plan succeeds if the robot ends up where it started, and fails otherwise
- The Script
  - Simulate responses to commands in the plan
  - First version: Every robot move succeeds
  - Second version: One or more moves fail

Run the Plan

- Start LUV (Lightweight UE Execution Viewer)
  - `luv`
- Run the plan/script from the command line
  - `run-ue -v -b <plan> <script>`
- Inspect node states, etc. in LUV