Model-based Prototyping of Real-time Systems With PapyrusRT and Unity

Michal Pasternak
Agenda

• Completed Work
  – Rover Simulation
  – Car Simulation

• Motivation

• Vision

• Work in Progress
  – Flow Chart
  – Components
- Simulations of both a Mars Rover and a self-driving Car.

- Communication which allows for full control of simulation through a generated papyrus model.

- Ability to hand customize the model, the environments, parameters and test conditions
Motivation

Key Motivations for a Prototyping Environment:

- Cost of hardware for testing is costly and often unavailable

- Most available simulators are not the best for the specified use case, do not allow the level of customizations required or their cost is prohibitive.

- Hand designing a specific simulation for testing a control model requires know-how and is time consuming.

- The required simulation changes frequently and requires frequent edits.
Vision

**Goal:**
Create a prototyping tool which will allow for the generation of a simulation environment and a control library for developing a control software using MDD.

**Requirements**
- Be able to customize simulation in a high level way, without having to edit any Unity code.
- Ability to customize the environment, model and simulation to conduct varied tests.
Flow Chart

- **Control Model**
  - **DSL to describe simulation**
  - **Simulation and Visualization**
  - **Control Model**

- **Generated Unity Scripts**
  - **Unity Simulation Framework**

- **User view**
  - **Socket Communication**

- **Uses**
  - **Creates**
  - **Generates**
• **Simulation Creator**: Uses a high level language to describe the environment, the controlled object, and what is to happen in the simulation.

• **Unity Simulation**: Generated based on criteria described in the Simulation Creator. Visual animation of object, environment and interactions. Produce relevant output from the simulation to the Papyrus Model.

• **Control Model**: The control model will take feedback from the simulation and provide it with inputs.