Model Driven Engineering, VT2015

Matthias Tichy, matthias.tichy@cse.gu.se
Grischa Liebel, grischa@chalmers.se
Fredrik Einarsson, freein@student.chalmers.se

Additional remarks for all assignment deliverables

- *State the authors of the deliverable (the group members)*
- *Correct language use, no grammar or spelling errors*
- *Reference and describe all figures/tables in the text*
- *Figures and graphs should be readable from a quality perspective*
- *Reference literature in your text where appropriate*
- *Ensure consistency between the different parts*
- *Define non-obvious acronyms*
- *The deliverable should be easily readable, understandable and complete*
- *Give arguments for your decisions (also using references)*
- *Show critical thinking*
- *Be prepared to get frustrated if something does not work as you think it should. Rise to the challenge!*

Project

Hard deadline for Handin via PingPong: 11.3. 23:59 (CET)

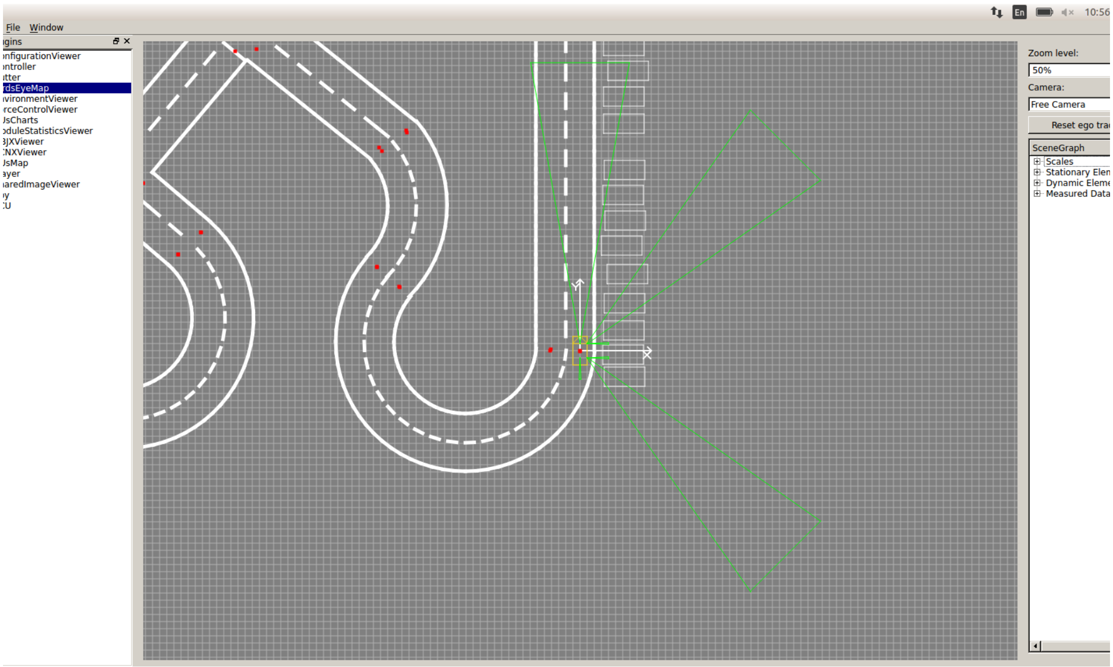
A. Introduction

The goal of the project assignment is to develop a modeling language for a self-driving car. As a self-driving car is obviously quite complex, we will focus only on the driving behavior, i.e., we focus on the actuators to accelerate, to brake, and to turn wheels, and different sensors which the car uses to sense its environment, i.e., radar sensors. Furthermore, we restrict the driving behavior to the specific case of finding a spot to park and park the car on a straight road.

We already provided you with a simulation environment for self-driving cars, which contains a simulation of the car including sensors and actuators as well as the environment. **However, the actual parking behavior is missing and**

should be generated from a model in the modeling language you develop in the project.

The following figure shows the simulation environment running. It shows a track as well as the car which is controlled by software (yellow box) and several parking cars (white boxes). The software-controlled car is turned facing “upwards”. And it has different sensors, which allow it to sense its surroundings (illustrated as green triangles).



Zoom level: 50%

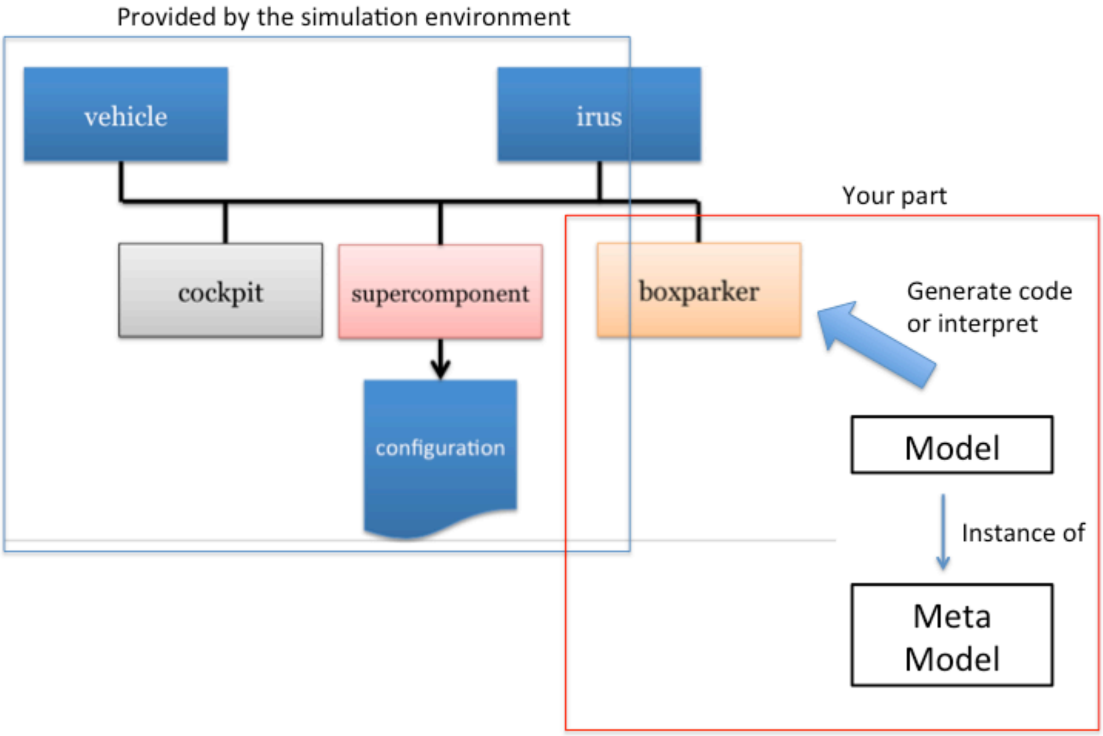
Camera: Free Camera

While the main assignment is the development of the modeling language and to generate the code for the parking behavior, it is strongly recommended that you first develop code for the parking behavior manually, and only thereafter building on the knowledge and experience gained develop the modeling language!

B. Develop a modeling language and tooling which allows to specify self-driving parking behavior

The main assignment is the development of a modeling language which allows the specification of self-driving parking behavior as a model (similar to the FSM modeling language in the lectures which enable the specification of different state machines). The model shall be integrated into the simulation environment as the BoxParker component and interact with the sensors and actuator interfaces provided. The integration of the model can be in different ways, e.g., generating source code out of the model and integrating the generated source code or generating a textual representation of the model and integrating a component which can directly interpret the model.

Or as a picture:



The modeling language shall be presented in a report for grading. The report should contain at least the following sections:

1. Introduction (max 1 page)
2. Overview of the solution (max 2 pages).
   This should give an overview about the concepts and ideas of your solution without going in the details of the meta model.
3. Metamodel: Ecore Metamodel and OCL constraints, (max 4 pages).
   Describe the purpose for each class, attribute and reference as well the purpose of the OCL constraints. Use examples where appropriate.
4. Concrete Syntax for the model (textual or graphical, both is not necessary) (max 3 pages).
   Present the mapping of the abstract syntax to the concrete syntax based on examples.
5. Model (instance of the meta-model)
   for the parking scenario in the concrete syntax you described in the previous section. Include a description how it works (max 3 pages)
6. Conclusion and Future Work (max 1 page)

- A short description of each group member listing his/her contribution (max 2 pages in total) (We might ask questions to check the contribution for the grading!).
- Please include a short (max 0.5 page) section, which lists where the main artifacts are found in the submitted ZIP-archive (e.g., metamodel, ocl constraints, model2text templates & generated code, model).
- Optional (and obviously ungraded):
  A short self-feedback and suggestions for next year's course (the project and its relation to the lectures) (max 1 page)

It is important that you not only show the different models and meta models. But also describe the different parts in detail and argue why you chose the presented solution (in comparison to alternatives), e.g., in the meta model you should explain why you choose the different classes and how they work together.

Furthermore, even as we restrict the scope of the model to the specific parking situation, the language itself should be designed in such a way that it is extendable to other driving situations. That means, a simple language which only allows to specify "Park" is obviously not enough.

Deliverable

- Document (pdf) reporting about the results for the project assignment based on the above presented list (12 font size and the above listed maximum pages, if one section is shorter you might use the saved space for another section.)
- A ZIP-archive of all plugins you developed/generated (including the model of your parking algorithm as well as the generated code.)
- A link to a downloadable virtual machine including a short tutorial how to open your model of the parking behavior, how to generate code for the parking simulation, and how to run the generated code in the simulation.