

A Domain-Specific Language for Lego Vehicles

Luc Engelen, Yaping Luo^{*}, Ana-Maria Sutii^{**}

^{*}y.luo2@tue.nl, ^{**}a.m.farcasi@tue.nl

October 7, 2014

1 Introduction

The goal of the assignments for this week and next week is developing a small domain-specific language (DSL) to control a number of Lego vehicles to simulate platooning movements. The idea of platooning is to link cars into a trainlike group, which can save fuel, fit more cars on the road, and potentially improve safety. In the group of cars, one car is leading the other ones and it is taking charge of the route to follow by the platoon. Therefore, the platooning that you will simulate will have one leading vehicle and a number of following vehicles. The platooning DSL will describe the vehicles taking part in the platoon, the route that these vehicles need to follow and constraints on the vehicles.

Besides the DSL for platooning, you will also be describing a DSL for a bounding box application. The bounding box application will just contain a list of routing commands for Lego Mindstorms.

You will then also produce NXC code for the platooning DSL and Java code for the bounding box DSL. A last step will be to make a transformation between a part of the platooning DSL to the bounding box DSL.

The exercise for this week will contain two assignments. The first assignment involves designing the metamodels for the two languages. The second assignment requires creating the Xtext grammar for the two languages.

For all the assignments, you will work with Eclipse and the Eclipse extensions Xtext, the Eclipse Modeling Framework (EMF), and the Epsilon toolkit. Installation instructions for Eclipse and its extensions are provided in a separate document.

2 First Assignment

The first assignment is meant to get you acquainted with the basics of EMF. You are asked to construct two Ecore models that serve as metamodels for the platooning language and the bounding box language. In the next assignment you will define the textual syntax for these two languages.

To complete this assignment, create two Ecore models following the tutorial on metamodeling that is provided as a separate document. Create an Ecore model named `BoundingBox.ecore` that represents the metamodel of the `BoundingBox` language, and an Ecore model named `Platoon.ecore` that represents the metamodel of the `Platoon` language.

Start by recreating the metamodel of the bounding box language, which is given in Figure 1. The metamodel defines moves along the X axis (left moves and right moves) and moves along the Y axis (upward moves and downward moves). Each of these moves is performed on a specified distance. To get an impression of the sort of programs you can write in this language, consider Figure 2.

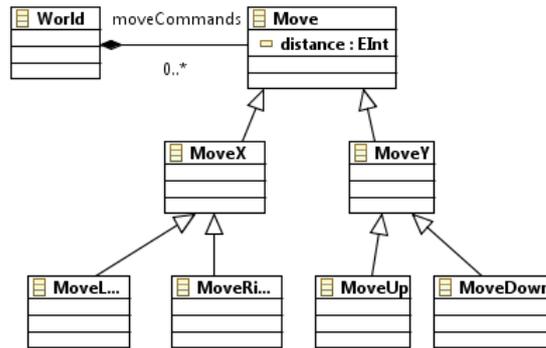


Figure 1: Metamodel for the BoundingBox language

Secondly, design the metamodel of the platooning DSL, which will allow you to write programs similar to the one in Figure 3. The platooning language should allow modelers to define three sections:

- the platoon;
- the route;
- constraints.

In the *platoon* section modelers specify a leading vehicle and the route that it follows. They also define a number of following vehicles, each of them with a front runner (i.e. the vehicle in front of it). In the *route* section modelers can write a list of commands that describe a route. There are two possible types of commands:

- Forward commands with a certain distance;
- Turning left or right (90 degrees).

Lastly, in the *constraints* section, modelers specify the headway (distance) between two vehicles in terms of a minimum and a maximum distance. An example of a complete specification can be found in Figure 3.

3 Second Assignment

For the second assignment, you must complete the grammars of the BoundingBox language and the Platoon language. Below, we provide an example of a BoundingBox model written in the BoundingBox language and an example of a Platoon model written in the Platoon language.

```

1 Box:
2   up 10
3   left 50
4   down 10
5   right 20
6   up 30

```

Figure 2: A BoundingBox model example in Xtext editor: The platoon will move inside the Box. You can assume that in the beginning the robot is facing north (the up direction). The up direction corresponds to the north, the left and right directions correspond to the west and the east respectively and the down direction corresponds to the south.

```

1 platoon:
2   LV Lego0 route R1
3   FV Lego1 front runner Lego0
4   FV Lego2 front runner Lego1
5
6 route R1:
7   forward(10)
8   turn Left
9   forward(50)
10  turn Right
11
12 constraints:
13  20 <= headway <= 30

```

Figure 3: A Platoon model example in Xtext editor: there is a leading vehicle called Lego0 and two following vehicles called Lego1 and Lego2 (lines 2-4). All the vehicles in the platoon follow the same route R1 (lines 6-10). The distance between vehicles is between 20cm and 30cm (lines 12-13)

The grammars in this assignment are defined using Xtext¹. A manual for Xtext and a number of tutorials are available on-line². You only need to look at Chapter 3 of the Xtext manual.

After completing a grammar, generate code from this grammar by executing the workflow that is located in the same folder as your grammar. Each time you execute this workflow, a metamodel based on your grammar will be generated. You can find this metamodel in the `model` folder of the plug-in for your grammar.

To complete this assignment, complete the grammars and ensure that the corresponding metamodels are the same as those created in the previous assignment.

¹<http://www.eclipse.org/Xtext/>

²<http://www.eclipse.org/Xtext/documentation.html>