

Assignment 3

2IMP20 - DSL Design

June 19, 2023

1 Introduction

In this assignment, you have to design and implement your own Domain-Specific Language (DSL). You are free to design and implement your own executable language, but it should be non-trivial and include concepts that have behaviour. Thus, markup languages (e.g., HTML and XML) are not allowed; neither are languages with only a few language constructs, like IFTTT. You may re-implement an existing language as long as you refer to the source where you found this language. You could also choose to implement an existing state machine language or some toy robot language. There are sufficient examples of DSLs you can choose from or get inspired by.

2 Assignment

The third assignment is meant to apply the concepts of the course to implement a DSL from scratch. The goal of the assignment is to build an executable DSL of your choice. You have to specify the syntax for the language; this involves concrete and abstract syntax. In addition, you have to define the static semantics of the language and its implementation. It is advisable to start with simple static semantic rules and gradually extend them. Finally, you must define a mapping from your DSL to a language that can be executed, for instance, C, C++, Java, or Python. This mapping can be implemented in different ways; choose a straightforward solution, for example, using templates (code generation).

You are free in the choice of your language workbench. In the first assignment, you used Rascal, but if you want to explore other language workbenches, such as EMF with Xtext and ETL/QVTo, MPS, or Spoofax, you can do this. However, be aware that other language workbenches have their own (steep) learning curve, take effort to get installed, and may lack proper documentation.

As part of the assignment, **each of you** is required to select and read three scientific papers related to the design/implementation of DSLs. These papers should provide context and better insight into the work you performed, and help you to reflect on it.

While the DSL design and implementation part of this assignment must be done in pairs, you are required to submit an individual report containing reflection on your experiences as a language designer, and discussion of the three scientific papers you read. It is advisable to each select and read your three papers early on, and to discuss them with your group partner to improve understanding.

3 Deliverable

For the third assignment, you must hand in the following deliverable:

- Individual report containing:
 1. A reflection on your experiences as a language designer. You must reflect on the use of the chosen LWB, the DSL you implemented, what you learned by designing and implementing a DSL.
 2. Discussion of three scientific papers you have selected and read related to the design/implementation of DSLs. These papers should provide you better insight into the work you performed.
- *You must work together with your partner to create the artifacts and documentation mentioned below.*
- All artifacts you have created, syntax definitions, static semantic rules and implementation, and code generators.
- Several non-trivial input programs or models and the corresponding generated outputs.
- Technical description/documentation of your language, including references to the sources you have used as inspiration, and a motivation for choosing this language, including the purpose of this language.
- Video where you show the language definition (both syntax and semantics), the creation of a program/model for this language, the transformation to source code, and the compilation and execution of the generated code.

4 Grading

The grading will be based on the following seven aspects:

- Individual reflection including discussion of 3 relevant scientific papers (4 pt)
- Description/documentation of the language (1 pt)
- Abstract and concrete syntax description (1 pt)
- Description of static semantic rules (1 pt)
- Implementation of the language (1 pt)
- Non-trivial input files and generated output (1 pt)
- Quality of the created video (1 pt)

5 Submission

You have to submit a single zip file containing all created files and the test programs, generated output documentation, video and (both) reflection report(s).

You have to submit this zip file *as a Canvas group of two students* before **12:00 noon of the 10th of July** via Canvas.