JOHANNES KEPLER
UNIVERSITY LINZ

Bachelor's Thesis

Coco/R Extension for Visual Studio Code:
Syntax highlighting and workflow-supporting features

Dipl.-Ing. Markus Weninger
Institute for System Software

P +43-732-2468-4361
F +43-732-2468-4345
markus.weninger@jku.at

Student: Byambabayar Altangerel
Advisor: Dipl.-Ing. Markus Weninger, BSc.
Start date: February 2021

Coco/R is a compiler generator, which takes an attributed grammar (written in Cocol/R) of a source language and generates a scanner and a parser for this language. Many projects use Coco/R to parse data, for example logs, configurations, and user input. It is also used to develop domain-specific languages.
While Cocol/R is relatively easy to learn and has a well-written user manual[1], it encompasses certain syntactical patterns that would profit from IDE support such as syntax highlighting or region collapsing. Such IDE support has been developed in the past for IDEs such as Eclipse or Netbeans.

The goal of this thesis is to develop an extension[2] for Visual Studio Code (VSC) to provide a better experience when working with Coco/R. VSC provides the possibility to add support for new programming languages[3], which can be splitted into *declarative language features* such as syntax highlighting, bracket autoclosing, auto indentation or snippet completion and *programmatic language features* such as auto completion or jump to definition operations. This thesis focuses on the first of the two.
In addition to implementing these declarative language features (which are performed using tokenization based on a TextMate grammar, which the student has to translate from the official Cocol/R EBNF), various workflow-supporting features should be implemented:

- A context menu to "Run" the currently opened .atg file (i.e., to generate a scanner and parser for the given attributed grammar)[4].
- The user should be able to specify workspace-specific settings - especially a output directory (i.e., where the scanner and parser files should be generated) and the package definition (i.e., in which package the generated scanner and parser should reside).
- Scanner.frame and Parser.frame files (needed by Coco/R during the generation process) should be automatically created if they do not exist yet.

Even though Coco/R can be used to generate scanners and parsers in various languages, the extension developed in this thesis only has to support Java. Java code should be formatted and highlighted according to Visual Studio Code's default Java settings (which is supported by VSC via *embedded languages*). Alignment of annotated Java code would be nice-to-have, but are not mandatory.

Modalities:
The progress of the project should be discussed at least every two weeks with the advisor. A time schedule and a milestone plan must be set up within the first 3 weeks. It should be continuously refined and monitored to make sure that the thesis will be completed in time. The final thesis must be submitted not later than 16.08.2021.

[1] https://ssw.jku.at/Research/Projects/Coco/Doc/UserManual.pdf
[2] https://code.visualstudio.com/api
[3] https://code.visualstudio.com/api/language-extensions/overview
[4] *Example Coco/R call: java -jar Coco.jar MyFile.atg -o /my/output/folder -package at.jku.coco*