Master Project Proposal Template

Paul Spencer: 11721677
University of Amsterdam
paul.spencer@student.uva.nl

January 2021

Project details

- **Project title:** Projecting Drools
- **Host organization:** Khonraad Software Engineering B.V.
- **Contact Person:** Toine Khonraad, MD, a.khonraad@khonraad.nl +31 (0)6 51 61 47 55

1 Project summary

Drools is an open source production rule system for complex event processing, using an implementation of the Rete algorithm. It has it's own Domain Specific Language, which are written in Drools files. When there are many rule in an environment it can be difficult to reason about the rules and how they interact. Although Drools is nearly 20 years old and has wide use, it does not have strong IDE support. This masters project will approach fixing this by creating a projectional editing IDE. There are two popular editing methods for language workbenches, these are free-form editing, where the user typically edits the source code, and projectional editing, where the user edits one or more projections of the persisted model. [1] Projectional editing is the method of bypassing the parser and programming directly into the Abstract Syntax Tree. In this project we will use the opensource language workbench MPS from JetBrains.

Khonraad Software Engineering's product, Khonraad, is a SaaS used by the Dutch local governments, implementing the Wvggz, Wzd and Wth laws. These laws deal with very sensitive details about mental health and domestic violence. It goes without saying that this data must not be accessed by the wrong people. Drools is the Technology choice Khonraad made to govern this critical function. Thus, they would like to improve the reasonability of this code.

This project will attempt to answer the following research questions:

- **RQ1:** How can we apply projectional editing to Drools to increase reasoning?

- **RQ2:** Which refactorings will most benefit a Drools code clarity?

2 Problem analysis

Here you present your analysis of the problem that your research will address. Also summarize existing scientific insight into the problem (result of your literature survey, see below). You may also touch on how this problem manifests itself at your host organization.

3 Research method

Present how you are going to find the answers to your research question. This section should cover:

- What will make the research difficult?
- What is the input you expect from the literature survey
- What sources will you use and how will you use / document them?
- What experiments / research will you do? What proof of concept will you make?
- What method will you use?
- Which hypothesis do you have?
- Present a time line
- How will you validate your research?

4 Expected results of the project

Give an explicit list of all the results that are expected from the project.

5 Required expertise for this project

Give a list of the expertise that is needed for the successful completion of this project. Next to each expertise, give your current level of proficiency. Examples are:

- Java programming
- Make and Makefiles
- Configuration management
- Testing

6 Timeline

An overview of what activity will take place when, and what milestones/deadlines the project has. You may want to illustrate the timeline using Gantt charts and/or PERT diagrams.

7 Risks

Give an assessment of the risks for this project and present contingency measures. These may include:

- Project goals are too ambitious.
- Project goals are too vague.
- The information about the project is unavailable/incomplete/too difficult.
- The gap between your expertise and the required expertise is too large

References

[1] Sebastian Erdweg et al. "The state of the art in language workbenches". In: *International Conference on Software Language Engineering*. Springer. 2013, pp. 197–217.