Master Project Proposal

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

January 2021

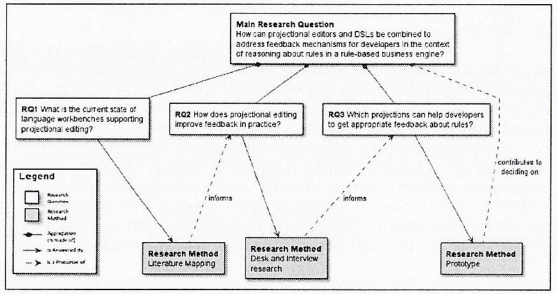
Project details

- **Project title:** Representation Mismatch Reduction for Development in Rules-Based Business Engines
- **Host organization:** Khonraad Software Engineering B.V.
- **Host supervisor:** Toine Khonraad, MD, a.khonraad@khonraad.nl

1 Project summary

In this project we will attempt to answer the research questions shown in figure 1

First explain the context and the problem


Main Research Question
How can projectional editors and DSLs be combined to address feedback mechanisms for developers in the context of reasoning about rules in a rule-based business engine?
RQ1 What is the current state of language workbenches supporting projectional editing?
RQ2 How does projectional editing improve feedback in practice?
RQ3 Which projections can help developers to get appropriate feedback about rules?
informs
informs
contributes to deciding on
Legend
Research Question
Research Method
Aggregation (is made of)
Is Answered By
Is a Precursor of
Research Method
Literature Mapping
Research Method
Desk and Interview research
Research Method
Prototype

make boxes on one level equal size, unless size carries information

Figure 1: Research Questions and Methods

A business rules engine execute rules at runtime. "Rules specify conditions to be monitored and operations that should be executed when certain conditions

* What problem does the algorithm solve?

** Following the less popular approach? Why?

*** This is engineering! Connection to research (questions)?

are detected. Rather than continuously monitoring the simulation, experts can define and deploy appropriate rules that are automatically evaluated at runtime"[14].

In our research we will be using the Drools rule Engine. Drools[1] is an open-source production rule system for complex event processing, using an implementation of the Rete algorithm[5]. It has a Domain Specific Language (DSL), in which rules are described. These are stored in Drools (.drl) files. Reasoning over a small number of rules is already surprisingly hard. Our host organization has many rules and, thus, reasoning about them is particularly challenging. This master's project will attempt to improve feedback whilst coding. This can reduce the representation impedance mismatch that hampers developer's reasoning.

*

Name it

A language workbench supports the efficient development of languages. The term caught on after a 2005 article by Martin Fowler[6]. Editing in language workbenches has two predominant editing forms - free-form text editing and projectional editing[4]. Free-form text editing is the more popular of these two. Projectional editing is a method of bypassing the need for a parser and programming directly into projections of the Abstract Syntax Tree.

Why?

We will create such an editor for the Drools DSL using a language workbench capable of creating projectional editors. On top of this newly modelled DSL, we will create new and different projections of the code for the purpose of increasing the reasonability of the code. To be clear, by reasonability we mean having the ability to reason.

*** **

you're the native speaker, but this does not sound elegant to me

For this project we will use the open-source language workbench Meta Programming System (MPS) from JetBrains[9]. MPS is built around the projectional editing paradigm. There is no existing implementation of the Drools language in MPS.

Although Drools is nearly 20 years old and has wide use, it does not have strong IDE support. One artefact of this master's project will be a prototype projectional editor, that will give much stronger editor support in IntelliJ, currently the most used Java IDE[10].

where?

2 Problem analysis

capitalisation?!

The mental health care and coercion laws in the Netherlands, Wvggz, Wzd, and Wth, provides agencies the ability to intervene in domestic violence, psychiatric disorders, and illnesses. Khonraad's system facilitates reporting and communication between municipalities, police, judiciary, lawyers, mental health care, and many social care institutions. The system has 15,000 users and is available 24/7. Configuration and administration use complex matrices of compliance mechanisms, access user rights and communication settings. The sensitivity of the personal data, being medical and criminal, means security is of utmost importance. The security against data loss, preventing unlawful disclosure and guaranteeing availability, especially during crisis situations, are crucial. Demonstration of the correctness of the, often changing, configuration is a major concern in the

Add a few sentences about what Khonraad SE BV is and what they do, where in the text fits best.

company.

In the current situation, configuration is done in a business rule system. This is Drools, a DSL from JBoss, a subsidiary of RedHat. Drools is a framework for Rule-Based development. The DSL is a textual representation of the abstractions of the rules. Currently it must be compiled to see if a set of rules are valid.

Editing programs in a text editor means that you must match the syntax for the parsers to transform the text into an AST. Projectional editors are editors in which a user edits the abstract syntax tree directly without using a parser[26]. This potentially allows for an almost unlimited language composition and flexible notations. Similar to the MVC Pattern, changes in one projection of the AST will instantly be visible and editable in another projection[8].

The problem of a lack of useful visualization for Drools has been known as far back as 2011, when Kaczor, et al[11] proposed a method of visualising Drools. There have also been a few commercial tools to help. ~~However, these all suffer from the parsing issue and lack of immediate feedback.~~ We are of the opinion that our approach will lend itself to a superior experience.

Be careful:
why should you succeed
where others failed?

3 Research method

The main question "How can projectional editors and DSLs be combined to address feedback mechanisms for developers in the context of reasoning about rules in a rule-based business engine?", will be answered by answering three sub questions.

Research question 1, "What is the current state of language workbenches supporting projectional editing?", will be answered by the method of conducting a literature mapping of the field of code reasonability measurement. This research method will follow the prescriptions of Kitchenham et al.[12].

Research question 2, "How does projectional editing improve feedback in practice?", will be answered by interviews with experts in the field of projectional editing, following the prescriptions of Mathers et al.[15].

how do you obtain
your interviewees?

right word?

Gregor[7], gives "A Taxonomy of Theory Types in Information Systems Research". For research question 3, "Which projections can help developers to get appropriate feedback about rules?", we will conduct what Gregor calls "Type V: Theory for Design and Action". The criteria for success of Type V research is that the prototype should "include utility to a community of users, the novelty of the artefact, and the persuasiveness of claims that it is effective". We intend for our prototype to meet these criteria.

who not....

~~We have observed~~ the difficulty that developers have trying to reason about and edit collections of Drools files. We hypothesize that developers can be presented with different views on their code that will allow them to better understand the code. The business problem we wish to solve - how to improve the ability to reason about large collections of Drools rules - appears to us to lend itself to the technique of projectional editing. Thus, we will apply projectional editing techniques, through the MPS language workbench to the

it is not clear to me why projectional editing
helps with the complexity issues of ~~the~~ large rule sets

Drools language. The novelty of our approach will be to create new view types specific to the needs of a Drools programmer.

We will be relying on MPS as well as other open-source components. The reason we chose MPS is that it is the most developed of the free and open source projectional editing language workbenches, found in a study of the state of the art in Language workbenches[4]. why not [3] as well?

Our designs of the projections, which will run in parallel to the Drools language modelling, will depend in part on the outcome of research carried out in the first period. Whether our design is appropriate with regards to performance and functionality is a risk. Whether we can achieve usefulness in our projections also presents a risk. We hope to mitigate this risk through literature review and academic supervision.

Risk management
deserves a separate
section → 7

The prototype will consist of a of the Drools language, re-defined in the MPS language. The prototype will further consist of a set of projections of the DSL's AST. MPS uses the Java graphics framework Swing for the creation of graphical, as opposed to textual, projections. During the building of the prototype we will decide upon which projections we will create. Some potential examples include:

- Visualization of order of rule execution.,
- Spreadsheet-like decision tables.,
- "Group-by" fact, query or function usage.

These are sentences!

The major tasks in this prototype development will be:

- Modelling the Drools language,
- Developing the alternative projections.

The prototype itself will be validated by working. However, if time permits, the hypothesis of the usefulness of the projections will be further validated through developer use surveys.

4 Expected results of the project.

We expect the following from this project:

- We will be able to model Drools in MPS.
- A suite of novel and useful projectional editors for the Drools language.
- We will reduce the thought to execution cycle for Drools Developers, resulting in a reduction of their "cognitive distance and representation impedance mismatch"[24].

A happy side effects of this project is that the following open-source products will become ~~generally~~ available: to the public (domain?)