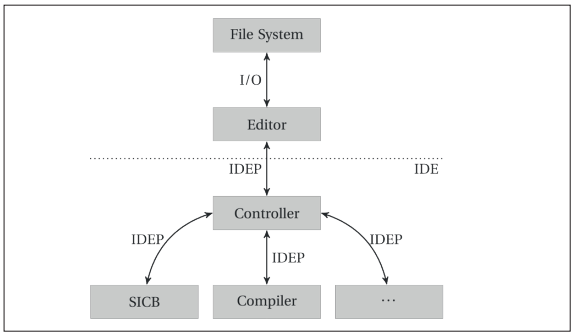
Cyrill
Schenkel

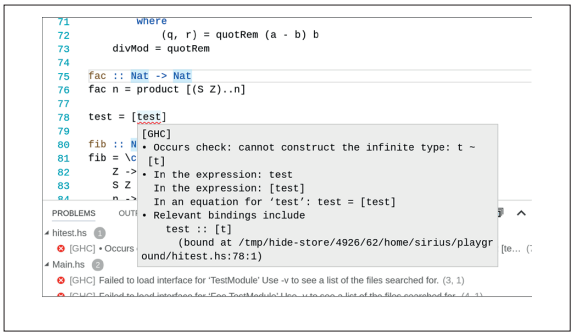
Graduate Candidate	Cyrill Schenkel
Examiner	Prof. Dr. Farhad D. Mehta
Co-Examiner	Dr. Simon Meier
Subject Area	Application Design

Modern IDE Support for Functional Programming

Proposal for an Architecture for the Next Generation of Functional Programming IDEs



Overview of architecture proposal

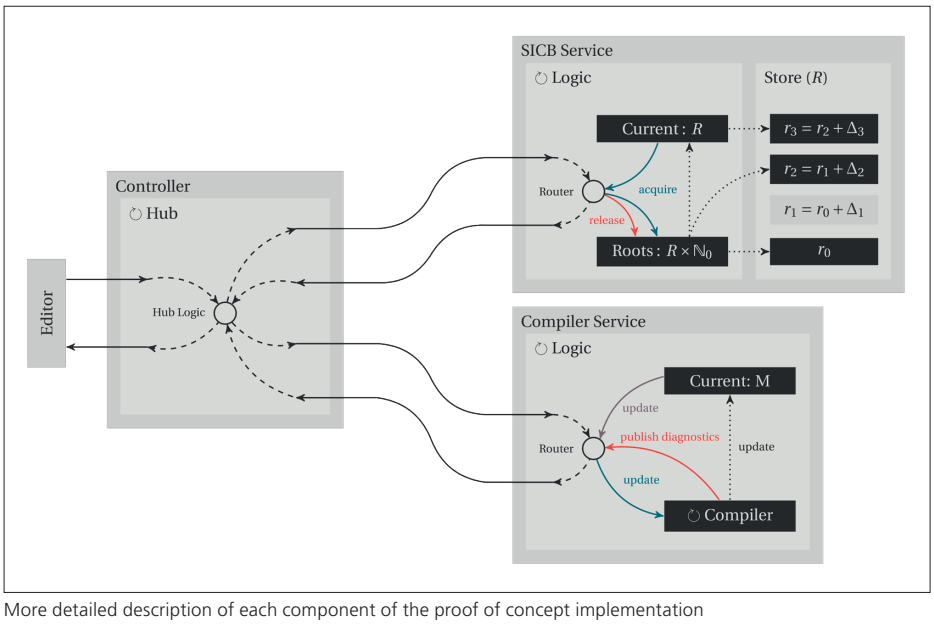


Overview of the proof of concept

Introduction: Functional programming concepts have recently received a lot of attention, as many mainstream programming languages, such as C#, C++ and Java, have adopted concepts like lambda expressions and lazy streams. Despite this development, the growth of adoption of functional programming languages by the industry still lags behind.

Approach/Technologies: By researching literature on this topic, interviewing students and industry professionals, who work with functional programming, and analysing the state of the art of functional programming IDEs, a set of requirements could be deduced. Based on this set of requirements an architecture for functional programming IDEs is proposed. Additionally, a proof of concept was implemented in Haskell to show how the results could be applied and what difficulties arise during the implementation of the proposed architecture.

Result: During the conceptual stage of the project, multiple architectural problems were found in existing functional programming IDEs. Based on the acquired set of requirements and these deficiencies, a new micro-services-based architecture is proposed along with a proof of concept implementation. There is still a lot of work to do in this area. Specifically in the case of Haskell, where the tools are very fragmented.



More detailed description of each component of the proof of concept implementation