

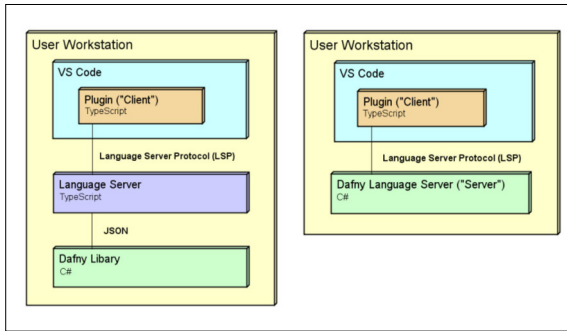
Marcel Hess



Thomas Kistler

Graduate Candidates	Marcel Hess, Thomas Kistler
Examiner	Thomas Corbat
Co-Examiner	Guido Zraggen, Google Switzerland, Zürich, ZH
Subject Area	Software Engineering - Core Systems

Enhancing Dafny Support in Visual Studio Code

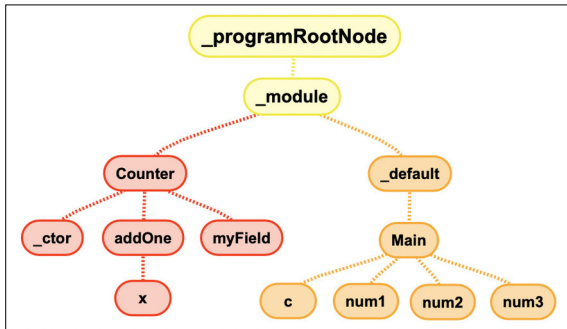


The language server was integrated into the Dafny backend in our preceding semester thesis. Own presentation

Initial Situation: Dafny is a formal programming language to proof a program's correctness. In a preceding bachelor thesis, a plugin for Visual Studio Code has been created to access Dafny-specific static analysis features. The plugin communicates with a language server using Microsoft's language server protocol, which standardizes communication between an integrated development environment (IDE) and a language server. The language server itself used to access the Dafny library, which features the backend of the Dafny language analysis, through a proprietary JSON-interface. In our preceding term project, the language server was integrated into the Dafny backend to render the JSON-interface obsolete.

Objective: This bachelor thesis continues the preceding term project and contains two major goals:

- The previously implemented prototype has to be improved in usability, stability and reliability.
- A symbol table has to be implemented to facilitate the development of symbol-related functionality like Rename or AutoCompletion. The symbol table is supposed to open a wide range for further development.



Result: A symbol table was created with an adaptive data structure. Every symbol contains information about its parent, its children and its declaration. This keeps navigation within the symbol hierarchy very simple. The implemented features GoToDefinition, Rename, CodeLens, HoverInformation and AutoCompletion benefit from the symbol table and are no longer required to analyze the code themselves and provide better results. The symbol table can be used for future extensions, such as AutoFormatting or CodeHighlighting. Beside features based on the symbol table, pre-existing functionality was revisited as well to improve the overall software quality. The final product has been deployed to the Visual Studio Code marketplace and is publicly available.

A symbol table has been implemented as a tree-based structured to represent all symbols inside the user code. Own presentation

The features HoverInformation and AutoCompletion benefit from the symbol table. Own presentation