Leveraging Generative Programming for Software Documentation

Mohammad Bilal, Dr. Jacques Carette, Dr. Spencer Smith
Department of Computing and Software, McMaster University, Hamilton, ON, Canada

Introduction

Objective: Generate **Software Requirements Specification** (SRS) in **mdBook** format from codified knowledge within **Drasil**, a **generative programming** framework.

What is mdBook? A command line tool for creating books in **Markdown**, converting content into styled **HTML**. [1]

Advantages of mdBook:
- **Navigation:** Sections are separated into individual pages.
- **Styling:** Advanced styling options for better readability and organization.
- **MathJax Support:** Includes LaTeX for mathematical equations and symbols. [1]

Motivation for Drasil

Problem:
- **Duplication:** Handwritten software artifacts often repeat information. [2]
- **Traceability:** Tracking origins and dependencies of software components is difficult. [3]
- **Maintainability:** Updating handwritten software is cumbersome and time-consuming. [3]

Solution:
- **Drasil:** a software framework written in **Haskell** that generates **all software artifacts** (requirements, design, code, tests, build scripts, documentation, etc.) based on a **single specification** in **a domain-specific language** (DSL). [3]
- Objectives: **Improves traceability** and **eliminates knowledge duplication**. [3]

Area of Focus:
- This research poster focuses on Drasil's **document generation** capabilities, specifically the Software Requirements Specification (SRS).
- Users define the SRS once in a **single location**, and Drasil generates the SRS in various output formats.
- Prior to the work done in this research poster, Drasil could generate documentation in **HTML**, **LaTeX**, and **Jupyter Notebook**.

Workflow

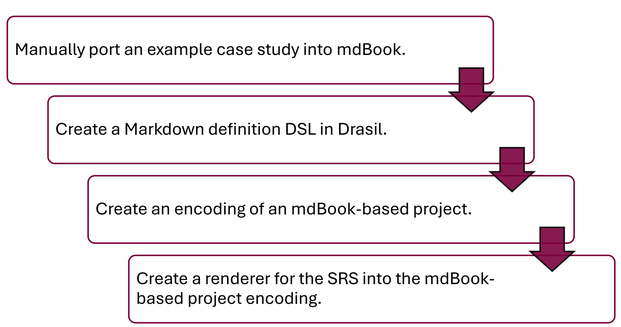

Figure 1: A flowchart detailing the steps followed to achieve the objective.

Results

Figure 2: A flowchart illustrating a simplified version of the document generation process in Drasil. The items highlighted in red represent the contributions from this research poster.

Discussion

- Adopting mdBook in Drasil enhances documentation quality and usability, aligning with goals of **improved traceability** and **eliminating duplication**.
  - Information is defined once and reused consistently across **all** documentation.
  - Maintaining and updating documentation is simplified, as editing the Drasil source code automatically updates all related documentation.
- With a Markdown definition DSL now established in Drasil, future work can focus on incorporating various Markdown flavors and generating new documentation formats.

Benefits of Generative Programming

Streamlined LaTeX updates: Updating the multiplication operator in LaTeX required just **a 4-line change in Drasil code**. This minor adjustment led to around **1,360 modifications** across **67 generated files**, highlighting Drasil's efficiency.

The graph below shows that software documentation generated with Drasil needed only **1,422 lines of user-written code**, producing **7,243 lines of generated code**—a **1:5** ratio!

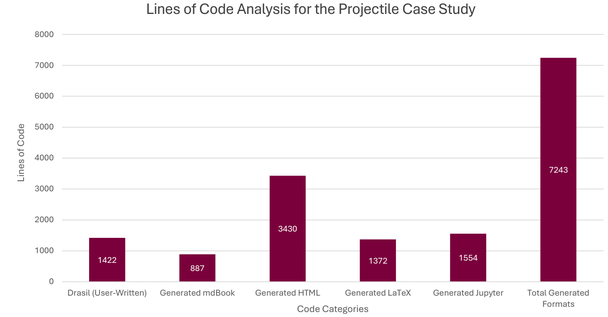

Figure 3: A comparison of user-written code vs. generated code lines.

References

[1] "MdBook documentation," Introduction - mdBook Documentation, https://rust-lang.github.io/mdBook/ (accessed Aug. 8, 2024).

[2] Carette, Jacques & Smith, Spencer & Balaci, Jason. (2023). Generating Software for Well-Understood Domains. 10.48550/arXiv.2302.00740.

[3] D. Szymczak, S. Smith and J. Carette, "POSITION PAPER: A Knowledge-Based Approach to Scientific Software Development," 2016 IEEE/ACM International Workshop on Software Engineering for Science (SE4Science), Austin, TX, USA, 2016, pp. 23-26.

Acknowledgements

I would like to express my gratitude to the Natural Sciences and Engineering Research Council of Canada (NSERC) for funding this position. I am also thankful to Dr. Spencer Smith and Dr. Jacques Carette for offering me this invaluable opportunity to learn and contribute to the Drasil project. Lastly, I extend my appreciation to Jason Balaci and Samuel Crawford for their support throughout the summer.