Leveraging Generative Programming for Software Documentation

Mohammad Bilal, Dr. Jacques Carette, Dr. Spencer Smith

Department of Computing and Software, McMaster University, Hamilton, ON, Canada

McMaster University | Computing & Software

Introduction

Problem:

- **Considerable Duplication:** Handwritten software artifacts often contain repeated information. [1]
- **Lack of traceability:** It is challenging to track the origins and dependencies of various software components. [2]
- **Lack of maintainability:** Maintaining and updating handwritten software is cumbersome and time-consuming. [2]

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). [2]
- Accomplishes two primary objectives: complete traceability and elimination of knowledge duplication. [2]

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**.

Objective

Goal:

- Expand Drasil's document generation capabilities by generating Software Requirement Specifications (SRS) in **mdBook**.

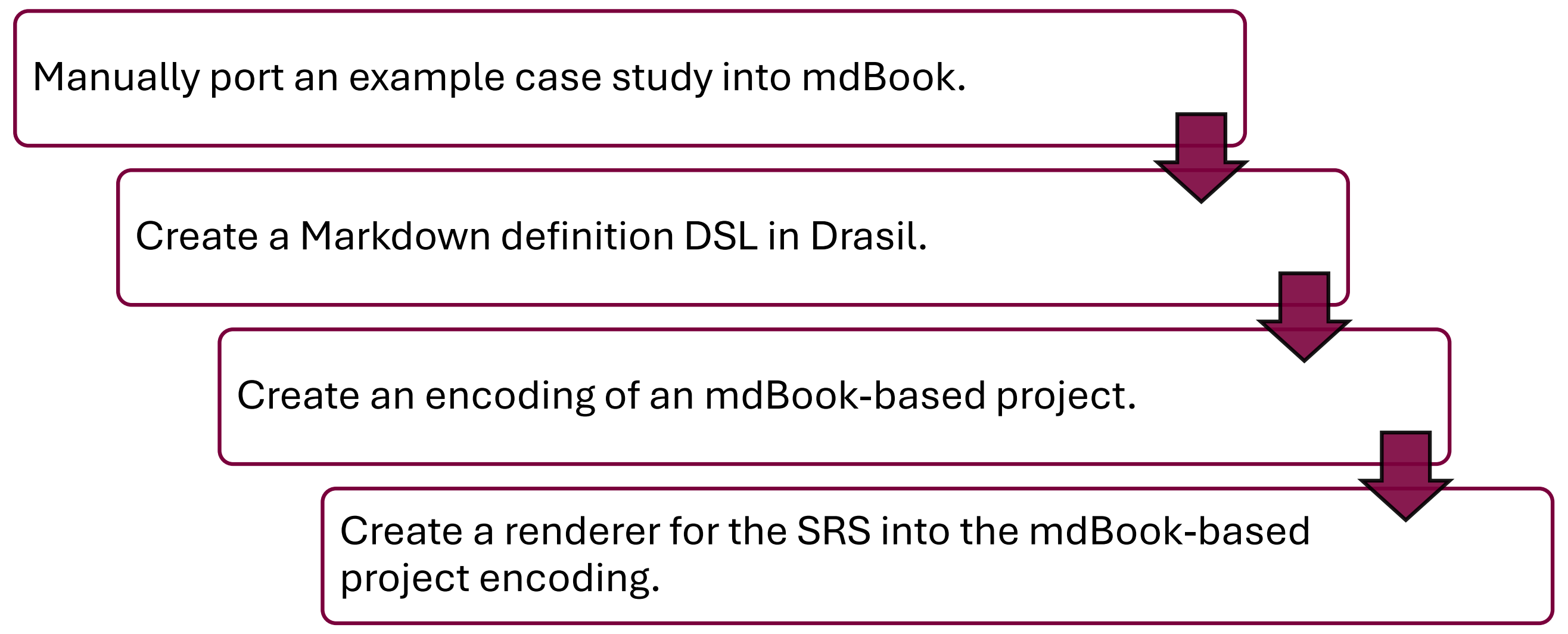
What is mdBook?

- mdBook is a command line tool designed for creating books in Markdown. [3]
- Content is authored in **Markdown**, and mdBook converts it into HTML, automatically applying styles and layout to enhance presentation.

Advantages of mdBook:

- **Improved Navigation:** Unlike the existing single-page formats, mdBook separates each section into its own page.
- **Enhanced Styling:** Offers advanced styling and layout options for better readability and organization.
- **MathJax Support:** Allows you to include mathematical equations and symbols in your documentation using LaTeX. [3]

Workflow



Manually port an example case study into mdBook.

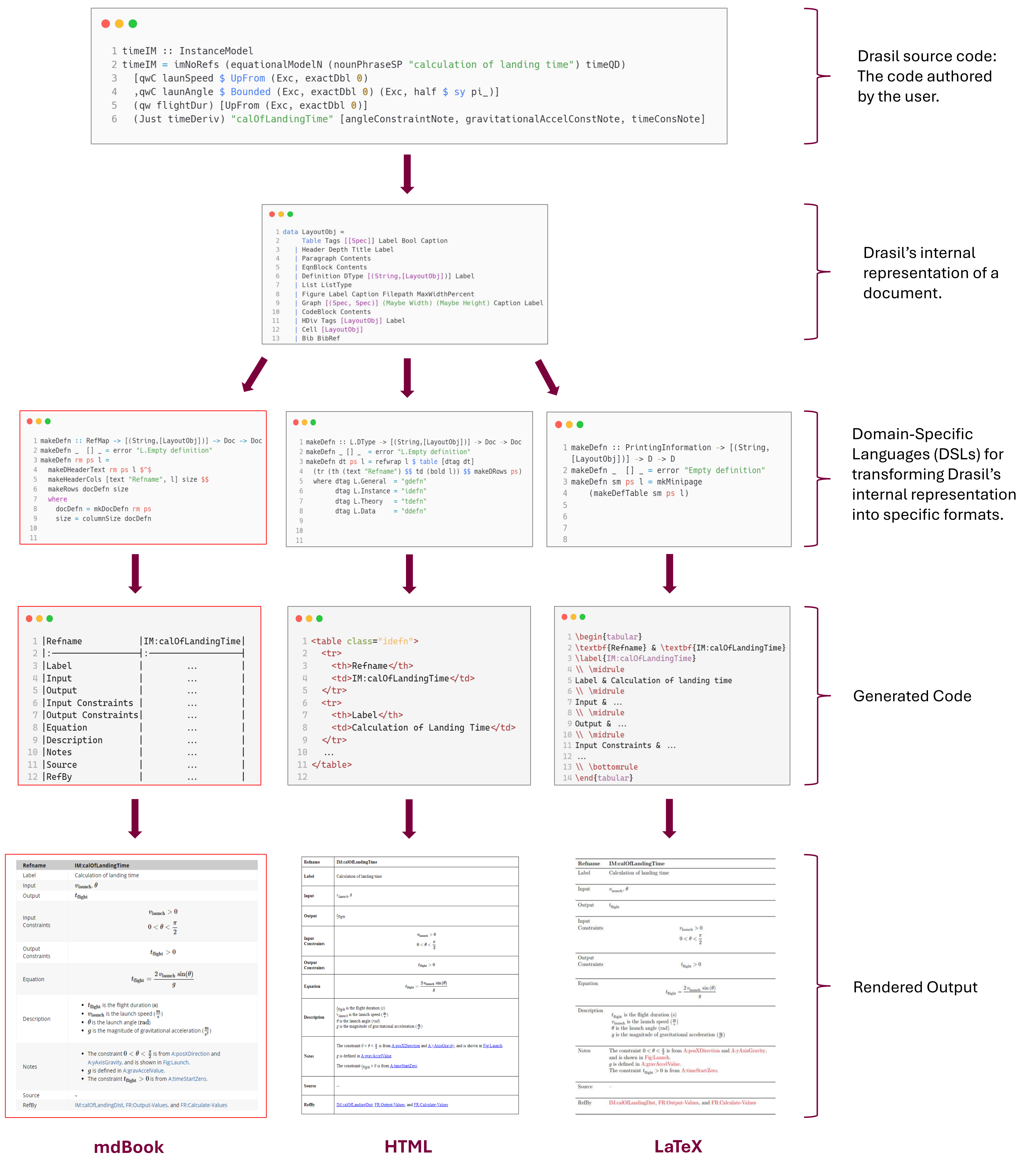
Create a Markdown definition DSL in Drasil.

Create an encoding of an mdBook-based project.

Create a renderer for the SRS into the mdBook-based project encoding.

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

Results



Drasil source code: The code authored by the user.

```haskell
timeIM :: InstanceModel
timeIM = imNoRefs (equationalModelN (nounPhraseSP "calculation of landing time") timeQD)
  [qwC launSpeed $ UpFrom (Exc, exactDbl 0)
  ,qwC launAngle $ Bounded (Exc, exactDbl 0) (Exc, half $ sy pi_)]
  (qw flightDur) [UpFrom (Exc, exactDbl 0)]
  (Just timeDeriv) "calOfLandingTime" [angleConstraintNote, gravitationalAccelConstNote, timeConsNote]
```

Drasil's internal representation of a document.

```haskell
data LayoutObj =
       Table Tags [[Spec]] Label Bool Caption
     | Header Depth Title Label
     | Paragraph Contents
     | EqnBlock Contents
     | Definition DType [(String,[LayoutObj])] Label
     | List ListType
     | Figure Label Caption Filepath MaxWidthPercent
     | Graph [(Spec, Spec)] (Maybe Width) (Maybe Height) Caption Label
     | CodeBlock Contents
     | HDiv Tags [LayoutObj] Label
     | Cell [LayoutObj]
     | Bib BibRef
```

Domain-Specific Languages (DSLs) for transforming Drasil's internal representation into specific formats.

```haskell
makeDefn :: RefMap -> [(String,[LayoutObj])] -> Doc -> Doc
makeDefn _  [] _ = error "L.Empty definition"
makeDefn rm ps l =
  makeDHeaderText rm ps l $^$
  makeHeaderCols [text "Refname", l] size $$
  makeRows docDefn size
  where
    docDefn = mkDocDefn rm ps
    size = columnSize docDefn
```

```haskell
makeDefn :: L.DType -> [(String,[LayoutObj])] -> Doc -> Doc
makeDefn _ [] _  = error "L.Empty definition"
makeDefn dt ps l = refwrap l $ table [dtag dt]
  (tr (th (text "Refname") $$ td (bold l)) $$ makeDRows ps)
  where dtag L.General  = "gdefn"
        dtag L.Instance = "idefn"
        dtag L.Theory   = "tdefn"
        dtag L.Data     = "ddefn"
```

```haskell
makeDefn :: PrintingInformation -> [(String,
  [LayoutObj])] -> D -> D
makeDefn _  [] _ = error "Empty definition"
makeDefn sm ps l = mkMinipage
    (makeDefTable sm ps l)
```

Generated Code

```markdown
|Refname            |IM:calOfLandingTime|
|:------------------|:------------------|
|Label              |        ...        |
|Input              |        ...        |
|Output             |        ...        |
|Input Constraints  |        ...        |
|Output Constraints|        ...        |
|Equation           |        ...        |
|Description        |        ...        |
|Notes              |        ...        |
|Source             |        ...        |
|RefBy              |        ...        |
```

```html
<table class="idefn">
  <tr>
    <th>Refname</th>
    <td>IM:calOfLandingTime</td>
  </tr>
  <tr>
    <th>Label</th>
    <td>Calculation of Landing Time</td>
  </tr>
  ...
</table>

```

```latex
\begin{tabular}
\textbf{Refname} & \textbf{IM:calOfLandingTime}
\label{IM:calOfLandingTime}
\\ \midrule
Label & Calculation of landing time
\\ \midrule
Input & ...
\\ \midrule
Output & ...
\\ \midrule
Input Constraints & ...
...
\\ \bottomrule
\end{tabular}
```

Rendered Output

Refname	IM:calOfLandingTime
Label	Calculation of landing time
Input	$v_{\text{launch}}, \theta$
Output	t_{flight}
Input Constraints	$v_{\text{launch}} > 0$ $0 < \theta < \frac{\pi}{2}$
Output Constraints	$t_{\text{flight}} > 0$
Equation	$t_{\text{flight}} = \frac{2\, v_{\text{launch}} \sin(\theta)}{g}$
Description	• t_{flight} is the flight duration (s) • v_{launch} is the launch speed ($\frac{\text{m}}{\text{s}}$) • θ is the launch angle (rad) • g is the magnitude of gravitational acceleration ($\frac{\text{m}}{\text{s}^2}$)
Notes	• The constraint $0 < \theta < \frac{\pi}{2}$ is from A:posXDirection and A:yAxisGravity, and is shown in Fig:Launch. • g is defined in A:gravAccelValue. • The constraint $t_{\text{flight}} > 0$ is from A:timeStartZero.
Source	–
RefBy	IM:calOfLandingDist, FR:Output-Values, and FR:Calculate-Values

mdBook

Refname	IM:calOfLandingTime
Label	Calculation of landing time
Input	v_{launch}, θ
Output	t_{flight}
Input Constraints	$v_{\text{launch}} > 0$ $0 < \theta < \frac{\pi}{2}$
Output Constraints	$t_{\text{flight}} > 0$
Equation	$t_{\text{flight}} = \frac{2\, v_{\text{launch}} \sin(\theta)}{g}$
Description	t_{flight} is the flight duration (s) v_{launch} is the launch speed ($\frac{m}{s}$) θ is the launch angle (rad) g is the magnitude of gravitational acceleration ($\frac{m}{s^2}$)
Notes	The constraint $0 < \theta < \frac{\pi}{2}$ is from A:posXDirection and A:yAxisGravity, and is shown in Fig:Launch. g is defined in A:gravAccelValue. The constraint $t_{flight} > 0$ is from A:timeStartZero.
Source	--
RefBy	IM:calOfLandingDist, FR:Output-Values, and FR:Calculate-Values

HTML

Refname	IM:calOfLandingTime
Label	Calculation of landing time
Input	$v_{\text{launch}}, \theta$
Output	t_{flight}
Input Constraints	$v_{\text{launch}} > 0$ $0 < \theta < \frac{\pi}{2}$
Output Constraints	$t_{\text{flight}} > 0$
Equation	$t_{\text{flight}} = \frac{2\, v_{\text{launch}} \sin(\theta)}{g}$
Description	t_{flight} is the flight duration (s) v_{launch} is the launch speed ($\frac{\text{m}}{\text{s}}$) θ is the launch angle (rad) g is the magnitude of gravitational acceleration ($\frac{\text{m}}{\text{s}^2}$)
Notes	The constraint $0 < \theta < \frac{\pi}{2}$ is from A:posXDirection and A:yAxisGravity, and is shown in Fig:Launch. g is defined in A:gravAccelValue. The constraint $t_{\text{flight}} > 0$ is from A:timeStartZero.
Source	–
RefBy	IM:calOfLandingDist, FR:Output-Values, and FR:Calculate-Values

LaTeX

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

- The adoption of mdBook for generating documents within Drasil not only enhances the quality and usability of the documentation but also aligns with the framework's goals of achieving complete traceability and eliminating knowledge 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: The rendering of the multiplication operator in LaTeX was updated, requiring only a **4-line change of Drasil code**. This small adjustment resulted in approximately **1360 modifications** across **67 files** of generated code, showcasing the efficiency of Drasil's generative approach.

In the graph below, software documentation generated using Drasil required only **1,422** lines of user-written code, resulting in **7,243** lines of generated code. That is approximately a **1:5** ratio!

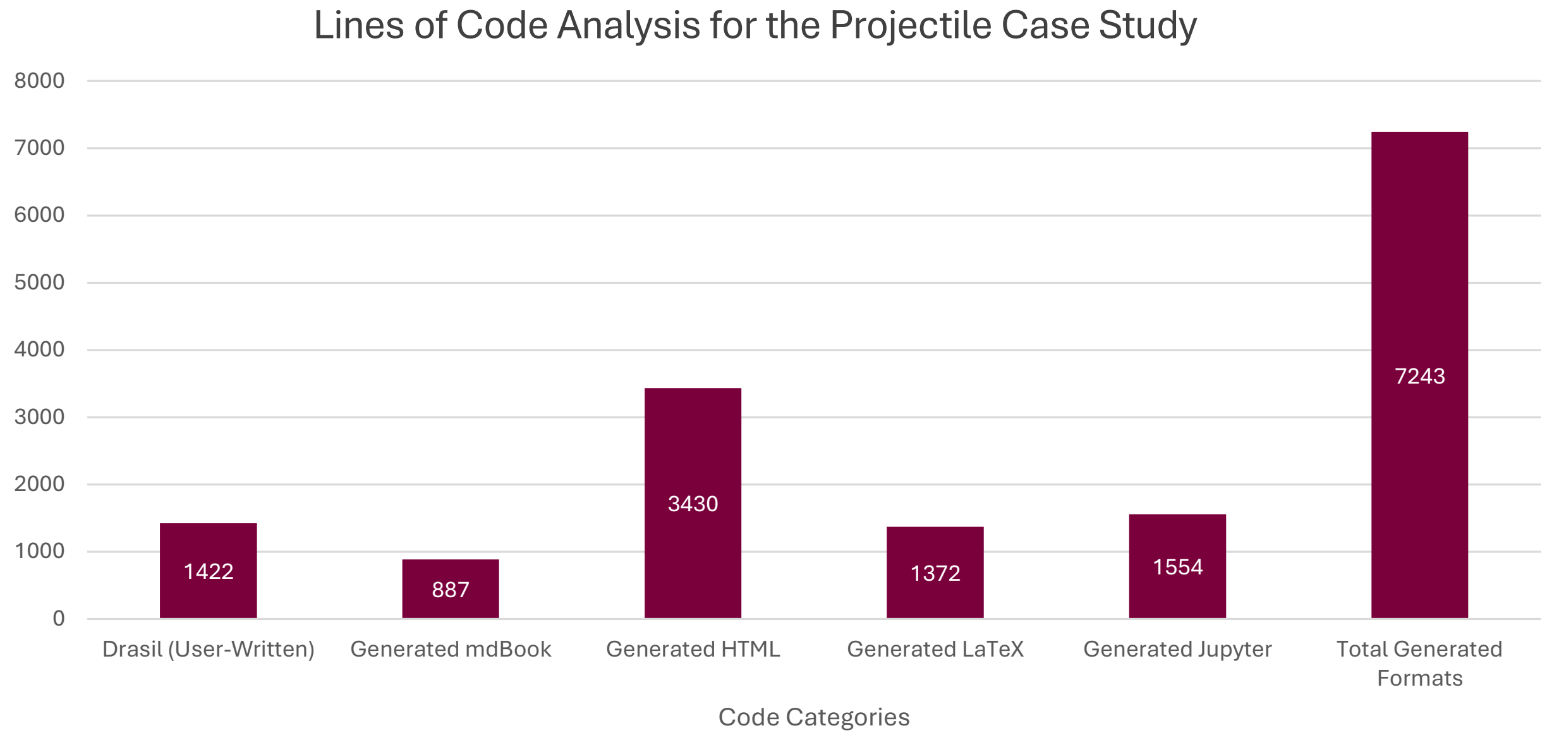


Figure 3: A comparison of user-written code vs. generated code lines.

References

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

[2] 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.

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

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.