Information Encoding and Traceability in Software

Drasil – Generate All the Things!

McMaster University
Computing & Software

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

Introduction

The Big Idea

- Software should be created by using a rational document-driven design with a focus on **communication**.[1]
- Using a structured design will **increase** developer **productivity**.[1]

The Problem

- **Maintenance** of information in software artifacts is difficult and time-consuming.[1]
- **Duplicate** information is prone to errors.[1]

The Solution

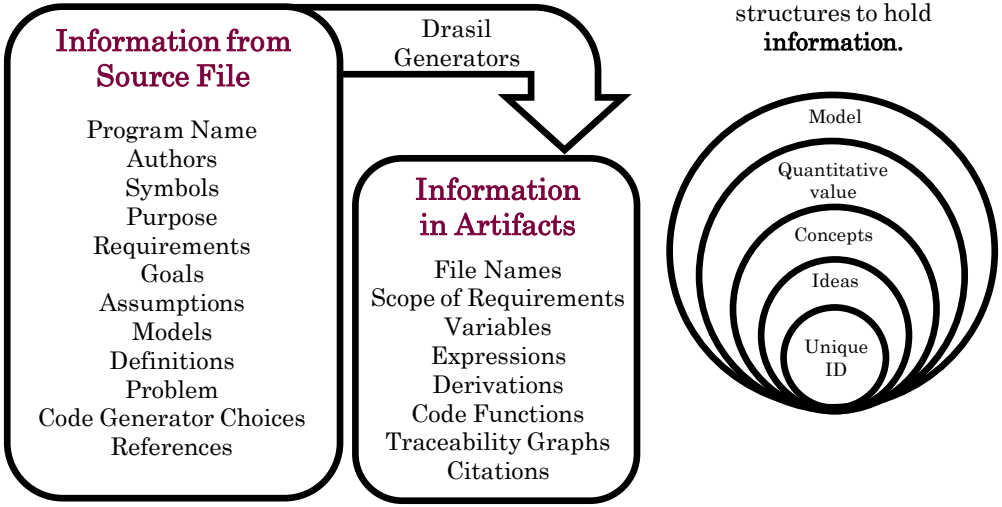
- **Drasil** can generate many software artifacts (documents, scripts, code, diagrams, etc.) from a **single source** of information.

Purpose/Objectives

- Improve the current Drasil framework.
- **Reduce** information **duplication** even more.
- Increase **traceability** of information.
- Demonstrate program flexibility.

Methods

Information Encoding – Teaching Drasil **information** with a semantic **meaning** makes that information **reusable**.



Chunks – Specialized data structures to hold **information.**

Recipes – Use chunks in a procedure to **generate documents** and code.

Results

- Easily generate and change documents.
- Errors were **pervasive** and much easier to fix.
- Low time investment needed to create new projects.
- **Flexible** enough to create other types of documents.
- Recipes rely on many embedded Domain-Specific Languages.

Conclusion

- Drasil is successful in making software **traceable** and **encoding information**.
- Creating relevant and up-to-date scientific documents is **efficient** and **easy**.
- Domains of knowledge in Drasil should be well-understood and maintained by experts.

Next Steps

- Automate the process of **gathering information** through advanced recipes.
- **Incorporate** information from other domains of knowledge.

Reference

[1] Daniel Szymczak, W. Spencer Smith, and Jacques Carette. Position paper: A knowledge-based approach to scientific software development. In *Proceedings of SE4Science'16 in conjunction with the International Conference on Software Engineering* (ICSE), Austin, Texas, United States, May 2016. In conjunction with ICSE 2016. 4 pp.

View our Work!

https://jacquescarette.github.io/Drasil

Acknowledgements

I would like to thank the Undergraduate Summer Research Award committee for funding this position and poster, Dr. Carette and Dr. Smith for giving me the opportunity to work on this project, and Jason Balaci for his help and support this summer.