DRASIL

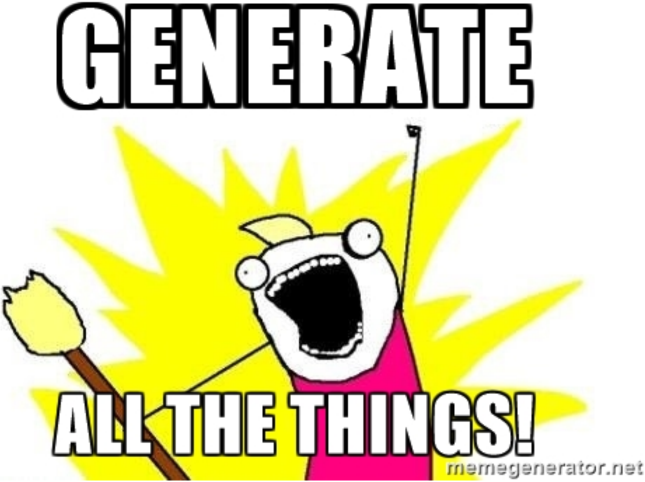
A Knowledge-Based Approach to Scientific Software Development

Aaron M, Dan S, Maryyam N, Nicholas R, Henry M

McMaster University

July 24, 2017

The Goal



Introduction

- Scientific and engineering computing has the potential to lead other fields of software with its solid knowledge base

Introduction

- Scientific and engineering computing has the potential to lead other fields of software with its solid knowledge base
- Drasil is intended to simplify the generation of documentation and code for scientific software

Introduction

- Scientific and engineering computing has the potential to lead other fields of software with its solid knowledge base
- Drasil is intended to simplify the generation of documentation and code for scientific software
- Also to facilitate desirable software qualities such as traceability, verifiability, and reproducibility

Introduction

- Scientific and engineering computing has the potential to lead other fields of software with its solid knowledge base
- Drasil is intended to simplify the generation of documentation and code for scientific software
- Also to facilitate desirable software qualities such as traceability, verifiability, and reproducibility
- case study from which structural patterns and implicit relationships can be extracted, data can be captured, and core systems can be tested and implemented

next page

- Patterns within examples; sentence combinators

next page

- Patterns within examples; sentence combinators
- Patterns between examples; extraction of common sections, contents, and concepts

next page

- Patterns within examples; sentence combinators
- Patterns between examples; extraction of common sections, contents, and concepts
- Knowledge extraction

next page

- Patterns within examples; sentence combinators
- Patterns between examples; extraction of common sections, contents, and concepts
- Knowledge extraction
- Reduce duplication

next page

- Patterns within examples; sentence combinators
- Patterns between examples; extraction of common sections, contents, and concepts
- Knowledge extraction
- Reduce duplication
- Implement new functions/types created by supervisors

next page

- Patterns within examples; sentence combinators
- Patterns between examples; extraction of common sections, contents, and concepts
- Knowledge extraction
- Reduce duplication
- Implement new functions/types created by supervisors
- Bug fixing

next page

- Patterns within examples; sentence combinators
- Patterns between examples; extraction of common sections, contents, and concepts
- Knowledge extraction
- Reduce duplication
- Implement new functions/types created by supervisors
- Bug fixing
- Opening/closing issues

Case Study Contributions

- SWHS
- NoPCM
- GlassBR
- HGHC
- SSP
- GamePhysics

end page

put content here