

Generate Everything] A Generic OO Language as a backend for a “Generate Everything” System

[

Jacques Carette^{a,*}, Brooks MacLachlan^b, Spencer Smith^a

^a*Computing and Software, McMaster University*

^b*Brooke's address*

Abstract

Abstract here

Keywords: code generation, document generation, knowledge capture,
software engineering, research software

1. Introduction

- extol the virtues of a “generate all things” approach - introduce Drasil -
requirements for Drasil to be effective - multiple back-ends, since research soft-
ware developers want code in their chosen language, has to fit into an ecosystem
5 of existing code - a design language that will allow for variabilities between the
generated code and documentation - generative back-end needs to be extensible
to new languages with minimal effort - the above requirements motivate the
design and implementation of GOOL

2. Background

10 - more on Drasil

3. Design of GOOL

- borrow from Brooks' MASc thesis document

*Corresponding author

4. Implementation of GOOL

- borrow from Brooks' MASc thesis document - maybe this would be where
15 we would mention the OO Patterns in GOOL (Chapter 6 of Brooks' thesis)?

5. Design Language

- some motivating examples

6. Literature Review

7. Future Work

20 **8. Conclusion**

References