

How I Did It!

H. Ackermann

Department of Computing Science
University of Aberdeen
Kings's College
Aberdeen AB24 3UE

Abstract

This report describes a revolutionary breakthrough in software engineering. An entirely new programming paradigm is presented in which the engineer writes a vague description of the requirements and the software system generates executable code using the *“I thought that’s what you meant principle.”* Artificial Intelligence techniques are used to sample fragments of existing code from commercial programs, and a low level parser is used to combine these fragments into an executable program. In 90% of cases, the given requirements are so vague that a panel of expert agree that the generated code could indeed be said to satisfy at least 70% of the requirement specification. Users of new the system found it to be almost indistinguishable from many existing desktop products. One expert remarked that “the new system could save many people thousands of dollars - they would never again need to buy expensive and often unnecessary software.”

Acknowledgements

I would like to thank my family, my colleagues, and everyone else who knows me.

Contents

1	Introduction (Why Its Important/Worthwhile)	3
1.1	Project Overview	3
1.2	Project Constraints	3
2	Background (Where I Started From)	4
2.1	Existing Code Generators	4
2.2	A New Paradigm for Code Generation	4
2.2.1	The Fractality Principle	4
2.2.2	Distinguishing Randomness and Complexity	4
2.2.3	Generating Random Code But Nice-Looking Code	4
3	Methods (How I Did It)	5
4	Results (How Good It Is)	6
5	Conclusions (What I Have Achieved)	7
5.0.4	Achievements	7
5.0.5	Future Work	7
5.0.6	Summary	7
A	An Appendix	8
	Bibliography	8

Chapter 1

Introduction (Why Its Important/Worthwhile)

In this report I will treat my grave subject with deadly seriousness. There may be doubters [?], but this document will demonstrate the error of their ways, and explain in full how I achieved my *remarkable results*.

The rest of this Report is divided into Chapters and SubSections. The Chapter Headings and some of the Subsection Headings might actually make sense. The rest of this stuff is of course complete rubbish.

1.1 Project Overview

1.2 Project Constraints

Chapter 2

Background (Where I Started From)

2.1 Existing Code Generators

2.2 A New Paradigm for Code Generation

2.2.1 The Fractality Principle

2.2.2 Distinguishing Randomness and Complexity

2.2.3 Generating Random Code But Nice-Looking Code

Chapter 3

Methods (How I Did It)

Chapter 4

Results (How Good It Is)

Chapter 5

Conclusions (What I Have Achieved)

5.0.4 Achievements

5.0.5 Future Work

5.0.6 Summary

Appendix A

An Appendix

Bibliography

- [1] G. N. Wilder and M. Brooks. Young frankenstein. *Variety*, 6(9):1236–1247, 1985.