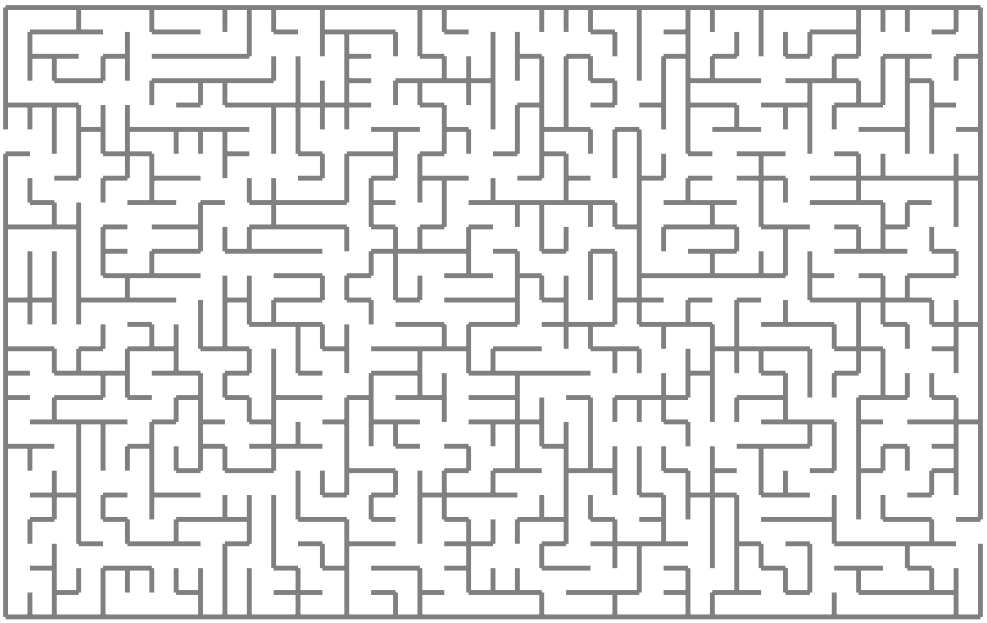
Formalizing WebDSL with Statix

Version of January 11, 2021



Max Machiel de Krieger

Formalizing WebDSL with Statix

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Max Machiel de Krieger
born in Delft, the Netherlands

Programming Languages Group
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

© 2021 Max Machiel de Krieger.

Cover picture: Random maze.

Formalizing WebDSL with Statix

Author: Max Machiel de Krieger
Student id: 4705483
Email: M.M.deKrieger@student.tudelft.nl

Abstract

WebDSL is a domain-specific language for web programming that is being used for over ten years. As web applications evolved over the past decade, so did WebDSL. A complete formal specification of WebDSL has been *TO-DO: check if missing or not updated* since its original development. With the introduction of Statix in the Spoofax language workbench, a declarative language that generates a typechecker, we made an elegant and practical formal semantics for WebDSL.

Thesis Committee:

Chair:	Prof. dr. E. Visser, Faculty EEMCS, TU Delft
Committee Member:	Dr. A. Katsifodimos, Faculty EEMCS, TU Delft
University Supervisor:	Ir. D. M. Groenewegen, Faculty EEMCS, TU Delft
Expert:	Ir. A. Zwaan, Faculty EEMCS, TU Delft

Preface

Preface here.

Max Machiel de Krieger
Delft, the Netherlands
January 11, 2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 An introduction to WebDSL

- Origins
- Key features
- Applications

1.2 WebDSL static semantics

- Product of research; changing frequently

1.2.1 Current implementation

- Spoofax introduction
- SDF3
- Stratego

1.2.2 Problems with current implementation

- Readability (scattered through files, dynamic rules)
- Maintainability (impact of changes can be unclear)

1.3 Statix

1.3.1 Introduction

- Origins
- Predecessors
- Examples

1.3.2 Why Statix will solve the problems

- Declarative and concise syntax that feels like formal specification rules

Chapter 2

Implementation

2.1 WebDSL in SDF3

2.2 WebDSL in Statix

Chapter 3

Evaluation

3.1 Correctness

- Defining correctness in absence of a formal specification
- How correct is the implementation WebDSL
- Explain correctness
- Edge cases

3.2 Validation

- How elegant in the Statix implementation?

3.3 Performance

- Explain metrics and methods
- Results
- Discuss results

3.4 Evaluating Statix

- Repeat reasons for using Statix
- What worked out as intended?
- What did not work as intended?
- What are the workarounds?
- Recommendations for improving Statix

Chapter 4

Related work

Chapter 5

Conclusion

Acronyms

AST abstract syntax tree

DSL domain-specific language

Appendix A

A