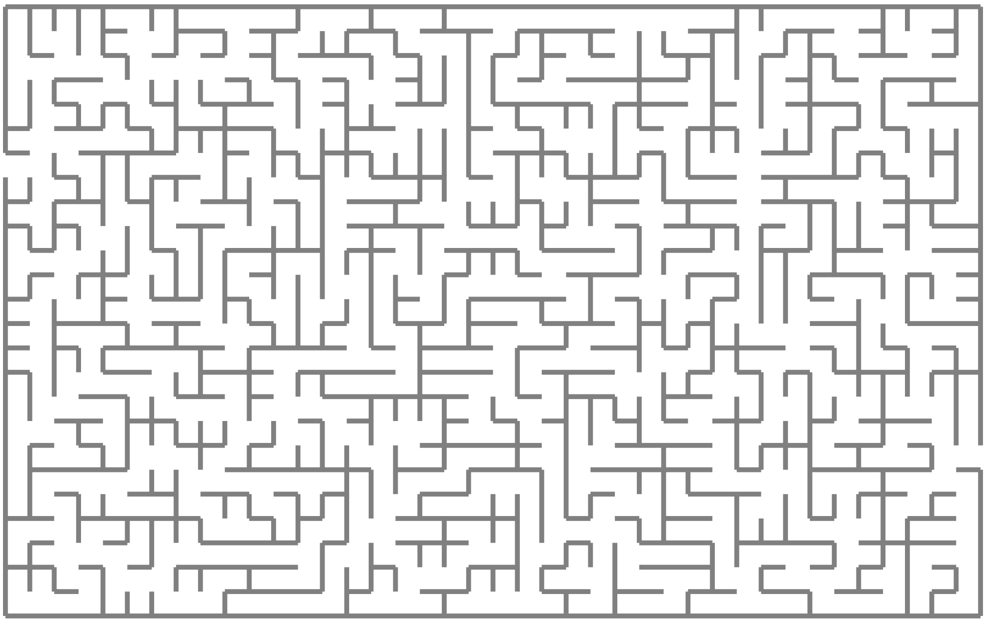
# Modernizing the WebDSL front-end: A case study in SDF3 and Statix

*Version of January 19, 2021*



Max Machiel de Krieger

# Modernizing the WebDSL front-end: A case study in SDF3 and 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

Cover picture: Random maze.

# Modernizing the WebDSL front-end: A case study in SDF3 and Statix

Author: Max Machiel de Krieger<br>
Student id: 4705483<br>
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 19, 2021

# Contents

**Preface** **iii**

**Contents** **v**

**List of Figures** **vii**

**List of Tables** **ix**

**1 Introduction** **1**

**2 WebDSL** **3**

**3 Concept** **5**

**4 WebDSL in SDF3** **7**

**5 WebDSL in Statix** **9**

**6 Evaluation** **11**

6.1 Correctness . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

6.2 Validation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

6.3 Performance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

6.4 Evaluating Statix . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

**7 Related work** **13**

**8 Conclusion** **15**

**Acronyms** **17**

**A A** **19**

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Research questions:

- Is it possible to capture the WebDSL static semantics with SDF3 and Statix?
- Is the implementation of the WebDSL static semantics in SDF3 and Statix performant?
- Is the WebDSL static semantics implemented in SDF3 and Statix elegant for the user?

# Chapter 2

# WebDSL

- What is WebDSL?
- How does WebDSL look like?
- How is WebDSL used?
- What non-trivial aspects does WebDSL have?
- How are the static semantics implemented now?
- Why change the implementation?

# Chapter 3

# Concept

- What does the solution look like?
- Why use SDF3?
- Why use Statix?

# Chapter 4

# WebDSL in SDF3

# Chapter 5

# WebDSL in Statix

# Chapter 6

# Evaluation

## 6.1 Correctness

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

## 6.2 Validation

- How elegant in the Statix implementation?

## 6.3 Performance

- Explain metrics and methods
- Results
- Discuss results

## 6.4 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 7

# Related work

- Papers about WebDSL
- Papers about modern Spoofax
- Papers about the definition of web programming languages
- Papers about modernizing (web) programming languages

# Chapter 8

# Conclusion

# Acronyms

| | |
|---|---|
| **AST** | abstract syntax tree |
| **DSL** | domain-specific language |

# Appendix A

A