To reuse or to be reused. Techniques for component composition and construction

de Jonge, M.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Contents

1 Introduction
  1.1 Software reuse ........................................ 1
  1.2 Abstraction ........................................... 3
  1.3 Component composition .................................. 5
  1.4 Component granularity .................................. 8
  1.5 Research questions
    1.5.1 Abstraction ...................................... 10
    1.5.2 Composition ..................................... 11
    1.5.3 Granularity ..................................... 11
  1.6 Overview ............................................. 12
  1.7 Origins of the chapters ................................. 14

I Development for Reuse ....................................... 15

2 Grammars as Contracts ....................................... 17
  2.1 Introduction .......................................... 17
  2.2 Concrete syntax definition and meta-tooling
    2.2.1 Concrete syntax definition ....................... 19
    2.2.2 Concrete meta-tooling ............................. 22
  2.3 Abstract syntax definition and meta-tooling
    2.3.1 Abstract syntax definition ..................... 23
    2.3.2 Abstract syntax tree representation ............. 24
    2.3.3 Abstract from concrete syntax .................. 25
  2.4 Generating library code ................................ 27
    2.4.1 Targeting C ...................................... 27
    2.4.2 Targeting Java .................................... 28
    2.4.3 Targeting Stratego ............................... 28
    2.4.4 Targeting Haskell ................................. 28
  2.5 A comprehensive architecture .......................... 29
    2.5.1 Grammar version management ...................... 29
    2.5.2 Connecting components ............................ 31
# Contents

## 2.6 Applications

## 2.7 Related work

## 2.8 Contributions

## 3 XT: a Bundle of Program Transformation Tools

### 3.1 Introduction

### 3.2 Program transformation scenarios

### 3.3 Transformation development

### 3.4 The XT bundle

### 3.5 Experience

### 3.6 Measuring software reuse

### 3.7 Concluding remarks

## 4 Pretty-Printing for Software Reengineering

### 4.1 Introduction

### 4.2 Pretty-printing

### 4.3 Pretty-printing for software reengineering

#### 4.3.1 Multiple output formats

#### 4.3.2 Layout preservation

#### 4.3.3 Comment preservation

#### 4.3.4 Customizability

#### 4.3.5 Modularity

### 4.4 GPP: a generic pretty-printer

#### 4.4.1 Format definition

#### 4.4.2 Format tree producers

#### 4.4.3 Format tree consumers

### 4.5 Applications

#### 4.5.1 The Grammar Base

#### 4.5.2 Integration of GPP and GB in XT

#### 4.5.3 A documentation generator for SDL

#### 4.5.4 Pretty-printing COBOL

### 4.6 Discussion

### 4.7 Concluding remarks

## II Development with Reuse

## 5 Cost-Effective Maintenance Tools for Proprietary Languages

### 5.1 Introduction

### 5.2 Language-centered software engineering

### 5.3 SDL grammar reengineering

#### 5.3.1 From YACC to SDF

#### 5.3.2 SDL grammar reengineering

### 5.4 An SDL documentation generator

### 5.5 Related work
III Epilogue

8 Conclusions
  8.1 Abstraction
  8.2 Composition
  8.3 Granularity
  8.4 Components and reuse

A Contributed Software Packages

B Additional Software Packages

Bibliography

Summary in Dutch / Samenvatting