

# Preface

Templates are used to generate all kinds of text, including computer code. The last decade, the use of templates gained a lot of popularity due to the increase of dynamic web applications. Templates are a tool for programmers, and implementations of template engines are most times based on practical experience rather than based on a theoretical background. This book reveals the mathematical background of templates and shows interesting findings for improving the practical use of templates. First, a framework to determine the necessary computational power for the template metalanguage is presented. The template metalanguage does not need to be Turing-complete to be useful. A non-Turing-complete metalanguage enforces separation of concerns between the view and model. Second, syntactical correctness of all languages of the templates and generated code is ensured. This includes the syntactical correctness of the template metalanguage and the output language. Third, case studies show that the achieved goals are applicable in practice. It is even shown that syntactical correctness helps to prevent cross-site scripting attacks in web applications. We apologize in advance for any errors that may have occurred in this text, and we would be grateful to receive any comments, or suggestions about improvements for further editions.

*B.J. Arnoldus*

AMSTERDAM, January 2012

*M.G.J. van den Brand, A. Serebrenik*

EINDHOVEN, January 2012

*J.J. Brunekreef*

UTRECHT, January 2012