AUTOMATION OF THE SERVER-SIDE DEVELOPMENT PROCESS FOR WEB APPLICATIONS

Valentin Mihai

Alecu Russo Bălţi State University, Bălţi, Republic of Moldova

mihaivalentin373@gmail.com

This research focuses on the comprehensive analysis of the server-side development process for web applications, encompassing testing and integration stages. Through this analysis, a well- defined software development scenario has been formalized. Additionally, the study examines the specific phases that can be automated, serving as the foundation for the subsequent theoretical basis of automation algorithms.

Moreover, contemporary architectural design patterns are investigated, specifically selecting patterns that adhere to the principles of automation. These selected patterns establish the fundamental templates, or stubs for the automation tool.

By conducting a comprehensive analysis of the development stages and patterns, it was possible to identify system components and processes that can be automated. The findings suggest that, given the provided specifications, a minimum of thirty percent of the code can be generated automatically.

Furthermore, this study delves into various approaches and paradigms associated with development automation. Metaprogramming and declarative programming paradigms are extensively examined, forming the basis for the meta-compiler algorithm and specification language. Consequently, precise requirements for the future compiler are formulated, the operational algorithm is formalized, and the specifications for the declaration language are derived.

Based on the aforementioned theoretical findings, a tool has been developed to automate the server-side development of web applications. As an experimental implementation, the Laravel framework has been employed. The implemented tool offers the following functionalities:

1. Processing of component specifications
2. Distribution of individual specification parts to child component compilers
3. Template population and source code generation for components.

The developed meta-compiler enables the automated creation of system components such as:

1. HTTP request descriptions
2. HTTP response descriptions
3. DTO object descriptions
4. Mapper descriptions

5. Model and migration descriptions
6. API documentation
7. Simple unit tests.

Consequently, the service layer (business logic rules) and the layer responsible for data interaction remain unaffected by these components. Therefore, the implemented meta-compiler automates the creation of eight out of the ten main components of the system, effectively streamlining the development process for the server-side of web applications.