

Editorial

The authors in the opening paper, “**Mapping Patterns Between XML and Annotations for Software Configuration**,” studied the mapping patterns between XML and annotations, focusing on their roles as metadata formats in software system configurations. The mapping patterns they studied include Direct Mapping, Nested Annotations, Enumeration, Wrapper, Distribution, Target, Parent, and Mixing Point—to facilitate the translation between XML and annotations. This study enhanced both formats based on specific configuration needs, thereby enhancing maintainability and reducing complexity in framework design.

In the second paper, “**A Scaffolding Tool for Generating Java Program Skeletons and Buggy Programs**,” the author provided a scaffolding tool designed to generate Java program skeletons and buggy programs from annotated solution code supplied by teachers. To create variations of programming exercises, the author used Java annotations and an annotation processor. This tool helps efficient teaching workflows and enhances student engagement through structured practice.

In the paper “**Role of Patterns in Automated Task-Driven Grammar Refactoring**,” the authors studied the design patterns used in automating task-driven grammar refactoring, focusing on how patterns can guide systematic transformations of grammars in software engineering. They proposed an approach that uses evolutionary algorithms to generate random refactoring operators. This approach enhanced the efficiency and reliability of support for semi-automatic grammar recovery and adaptation, particularly in complex scenarios such as removing left recursion or adapting legacy grammars.

We hope that these papers generate further interest in the directions set forth.

Editors