



# Recommendation of language patterns during language design

Ing. Katarína Šipošová | Supervisor: Ing. Sergej Chodarev PhD. | Faculty of Electrical Engineering and Informatics, Technical University of Košice

## Motivation and Goals

**Language pattern** is a mapping between abstract and concrete syntax in a formal language that ensures readability and unambiguity of the language [1].

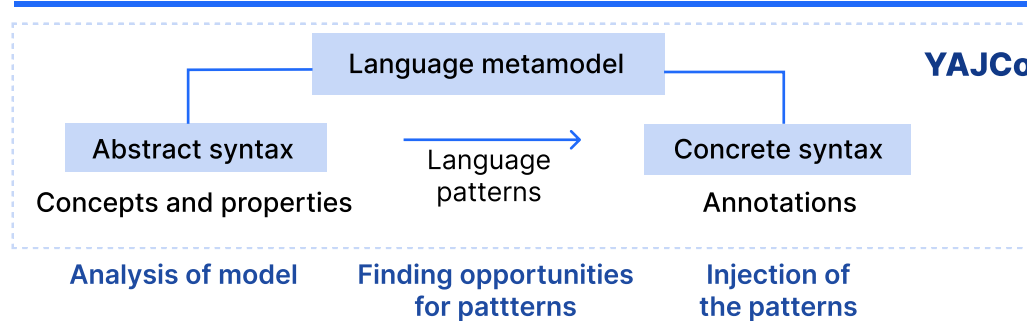
In the abstract-syntax-first approach to the language design, where the metamodel is the resort of the design, language patterns define typical forms of notation, such as keywords, separators, or infix operators.

Our goal is to provide support of the patterns by automated generation of pattern suggestions, thus supporting clear and readable syntax design. We aim to analyse a language metamodel based on predefined heuristics in order to identify possible pattern utilizations.

## YAJCo

YAJCo is a framework for the abstract-syntax-first language definition. It allows to define a language concepts as Java classes and concrete syntax using Java annotations [2]. We used YAJCo as an implementation of the language metamodel definition.

## Proposed solution



Language design process starts with the definition of the language model — its concepts and their properties. Usually, the process would continue with the definition of the concrete syntax — the designer would select and apply suitable language patterns.

We propose a way in which a language model is analysed based on defined heuristics. These heuristics come from the patterns definitions. The analysis is automated. However, it is not expected to be fully independent and might need some input from the language designer. After model analysis, the relevant patterns are proposed to the author. When a pattern suggestion is accepted, it is injected into the model by adding annotations to the concept definition.

## Pattern tool

The realisation of the proposed solution is the Pattern tool. It is a command line tool that, given a language model, analyses the language and suggests language patterns to be used in the language. The model the Pattern analyses is serialized YAJCo language model. Pattern can be launched in the interactive mode that allows the language designer to aid the analysis with additional information, to provide more precise pattern suggestions, especially in the cases when the analysis cannot decide on its own. Pattern can in addition manipulate the Java classes defining language model and thus implement the patterns to the language by adding YAJCo annotations.

## Evaluation and results

The evaluation was performed on the selected YAJCo language models with removed patterns representing concrete syntax. Pattern tool was used to perform analysis of such models. The produced pattern suggestions were compared with the original pattern definitions and the relevance of the suggestions was evaluated. The values of the precision (P) and recall (R) of the analysis were calculated based on the true positive (TP), false positive (FP) and false negative (FN) findings.

Language	TP	FP	FN	P	R
JSON	7	3	3	0.70	0.70
StateMachine	6	3	4	0.67	0.60
DeskNielsen	3	1	4	0.75	0.43
MathExpression	10	0	0	1.00	1.00
simpleRobot	2	2	2	0.50	0.50

## Conclusion

We showed that it is possible to suggest a language patterns based on the language model analysis. However, as it was anticipated, the analysis is not fully independent and needs some additional input from the designer. This was considered in our Pattern tool implementation, that supports interaction. The automated pattern recommendation can support language designer in the concrete syntax design decisions and therefore provide more readable syntax.

## References

- [1] S. Chodarev, J. Porubän, J. Juhár, M. Sulír, M. Bačíková, "Language Syntax Design Patterns" — article in progress
- [2] J. Porubän, M. Forgáč, and M. Sabo, "Annotation based parser generator," in 2009 International Multiconference on Computer Science and Information Technology. IEEE, 2009, pp. 707-714.