Recurring Return on Modeling Investment: A Conceptual Modeling Language and Extensible Compiler

Quenio Cesar Machado dos Santos[1] and Raul Sidnei Wazlawick[2]

[1] Computer Sciences,
UFSC - Universidade Federal de Santa Catarina, Brazil,
`queniodossantos@gmail.com`
[2] Associate Professor of Computer Sciences Department,
UFSC - Universidade Federal de Santa Catarina, Brazil,
`raul@inf.ufsc.br`

Abstract. Proposes a textual programming language that enables conceptual modeling (similarly to UML classes/associations and OCL constraints) and a compiler that allows code generation (via extensible textual templates) to any target language or technology. Together, the language and the compiler make it feasible to specify (in a single high-level language) the information of ever-changing, increasingly distributed software systems. From this single source, the automated code generation keeps the implementations (across the different platforms and technologies) consistent with the specification. Also, as the technology landscape evolves, these textual models allow the recurring use of the investment made on their specification. Unlike other approaches, such as MDA and MPS, the built-in tooling support, along with the textual nature of this programming language and its extensible templates, facilitates the integration to the workflow of software developers, which is expected to promote its adoption.

Keywords: conceptual modeling, UML, OCL, MDA, MPS, programming language, compiler, code generation, model-driven software development, modeling investment, classes, associations, constraints, specification, software tools, metaprogramming, generative programming

References