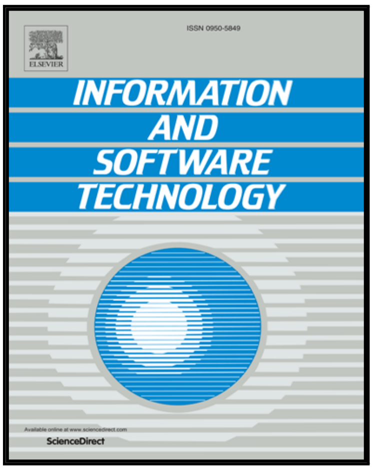

Object Constraint Language for Code Generation from Activity Models

E.V. Sunitha , Philip Samuel

PII: S0950-5849(16)30419-0
DOI: 10.1016/j.infsof.2018.06.010
Reference: INFSOF 6008

To appear in: *Information and Software Technology*

Received date: 8 December 2016
Revised date: 31 May 2018
Accepted date: 21 June 2018

Please cite this article as: E.V. Sunitha , Philip Samuel , Object Constraint Language for Code Generation from Activity Models, *Information and Software Technology* (2018), doi: 10.1016/j.infsof.2018.06.010

Object Constraint Language for Code Generation from Activity Models

Sunitha E.V
Department of Computer Science, Cochin University of Science and Technology, Kochi – 682022
sunithaev@gmail.com

Philip Samuel
Department of Computer Science, Cochin University of Science and Technology, Kochi – 682022
philips@cusat.ac.in

ABSTRACT

Context : Achieving hundred percent automation in code generation process from Unified Modeling Language(UML) models will make a drastic advancement in software industry. UML does not use a fully formalized semantics. So it leads to ambiguity during automatic implementation of UML models. These ambiguities can be avoided to a large extent using Object Constraint Language (OCL). OCL is formal and user friendly which is also familiar to industry people.

Objective : This paper examines how to improve the code generation from UML models, with the help of Object Constraint Language. It also explores the possibilities to incorporate OCL in UML activity models and generate code from the OCL enhanced activity diagrams.

Method : Meta models for the association of OCL expressions with the UML activity diagram is proposed in the paper. OCL expressions are added as part of the UML activity models to improve the code generation and to specify assertions and behavior. Moreover a tool, called ActivityOCLKode, is implemented which follows the algorithm for code generation. The algorithm is depicted in the text.

Results : The tool which is implemented based on the proposed method gives a promising result. More than 80% of source code is generated using the tool. In addition, the average execution time for our approach is only 11.46 milliseconds.

Conclusion : The meta model proposed in the paper gives the strong theoretical back ground to attach OCL statements with each element in the UML activity diagrams. The proposed method of code generation will improve the productivity of the software industries, since it reduces the software development effort and time. Since UML and OCL are commonly used in software industry, our method is easily adaptable by software programmers in industry.

***Key words*— Code generation, UML, XML, OCL, activity diagram**

I. INTRODUCTION

UML (Unified Modeling Language) is a standard used for software designing [4]. Activity diagram is a behavioral diagram in UML which is used to model work flow and object flow in a system. It includes elements to show control flow. For example, action, activity, activity edges, swim lanes (partitions) etc, are some of the strong elements in activity diagram. The activity diagram can be considered as a graph [29]. The activity graph contains nodes and edges. The nodes can be control nodes such as initial & final nodes, decision nodes, fork node, join node etc. Edges in activity graph are the activity edge which shows the transition from one activity to another. These nodes can be converted to programming constructs without much complexity. The edges give the sequence order of the operations (or activities). The concept of activity graph helps us to traverse through the activity diagram and generate the overall execution logic of the system.

UML is not a fully formal language. Its semantics are not fully formalized. In many places natural language is used for model specification. It leads us to a scenario where the precise model presentation is difficult. So, whenever we use activity diagram, or any UML diagram, for code generation, it is recommended to complement it with specification languages like Object Constraint Language (OCL) [22, 30, 41]. OCL can supplement some of the shortcomings of UML notations, like lack of precision. We need to give the constraints of the objects in the UML models. Usually, these constraints are given in natural languages. This may be ambiguous. A formal language is required to unambiguously present the constraints. Since OCL is a formal language, and all constructs in OCL are well defined, it can unambiguously specify the constraints on the object or system. At the same time OCL is familiar and widely used in software industry.

OCL expressions are used with UML models to formally specify the constraints in a precise and concise way, where the graphical notations fail to do so. It can be used for specifying pre/post conditions on operations and methods, the actual parameters that are passed to the operations, the initial values of the attributes, the guard conditions etc. It can also be used to specify invariants and types in UML class diagrams. It can be even used as navigation language and also to specify the well formedness rules of the UML meta model.