Automated Assessment of UML Activity Diagrams

Michael Striewe, Michael Goedicke
Paluno – The Ruhr Institute for Software Technology
University of Duisburg-Essen
Essen, Germany
michael.striewe@paluno.uni-due.de, michael.goedicke@paluno.uni-due.de

ABSTRACT
Current approaches to automated assessment of UML diagram mainly focus on static analysis. As this may be insufficient for behavioural diagrams, this contribution presents an approach to dynamic checks of UML Activity Diagrams. The approach makes use of trace generation and sequence alignment and shows fair results in preliminary experiments.

Categories and Subject Descriptors
K.3.1 [Computers and Education]: Computer Uses in Education—Computer-assisted instruction (CAI)

General Terms
Design, Languages

Keywords
Modeling; E-Assessment; UML Activity Diagrams; Feedback Generation

1. INTRODUCTION
Introductory courses to system modeling using the UML are a typical part of computer science curricula. If many students attend these courses, detailed feedback to solutions of modeling exercises cannot be provided with short response times. To help in this problem, some approaches exist that enable automated assessment of particular diagram types of the UML [1, 3] or static analysis of UML diagrams in general [4, 2]. However, the behavioural diagrams of the UML do not only have static properties, but also dynamic semantics. As with programming exercises, which are typically checked both statically and dynamically, these kinds of diagrams should also be checked in two ways. Hence this contribution presents work in progress towards fully automated checks of UML behavioural diagrams using UML Activity Diagrams as sample instance.

2. TRACE-BASED CHECKING
Checking diagrams dynamically means to compare the semantics of the solution to reference semantics. The semantics of UML Activity Diagrams are in general defined by token flows: Tokens represent objects, dates, or control and are passed from one activity to another via the edges in the diagram. The sequence of activities that accept a token on its way from its creation to a final node can be considered as a trace. Forks may occur that cause several tokens to be passed around in parallel. The resulting trace thus contains parallel sequences that need to be reduced to sequences deterministically before traces can be compared. Then the solution trace and the reference trace can be aligned using algorithms from sequence alignment as used in bioinformatics, e.g. Ukkonen’s algorithm [5]. This algorithm uses a scoring function that adds positive scores for matches of sequence elements, and subtracts scores for mismatches or gaps. Based on the individual alignments an overall match score for all sequences generated by one solution can be computed and used for feedback generation.

3. PRELIMINARY RESULTS
Experiments with solutions from a bachelor degree course on UML modelling showed satisfying results with the feedback quality. In particular, the approach was able to accept solutions using alternate modeling approaches (such as using interrupt sections instead of decision nodes) without problems. However, the approach still shows some weaknesses with respect to mismatches of activity names.

4. REFERENCES