Automated Checks on UML Diagrams

Michael Striewe, Michael Goedicke
University of Duisburg-Essen
Motivation

- Modeling with UML is topic in introductory courses
  - Mass validation of exercises is tedious task
  - Benefits for mass validation known from e.g. programming exercises
- Known approach:
  - Check conformance to a sample solution
  - Drawbacks: Ambigious, correctness vs. quality
- Our approach:
  - Rule based checks
  - Inspired by teacher’s way of working
Example

- Small exercise: Customers of phone companies and their invoices
- Things to check:
  - Structure with respect to the exercise
  - Names with respect to the exercise
  - General style issues
Technical Background

• Handling solutions
  • Using XMI 2 as exchange data format
  • Parsing XML to syntax graph

• Handling rules
  • Using graph query language
  • Adding different interpretations for desired / undesired elements
    • „matching query“ vs. „matching rule“
  • Adding points / grades
  • Adding feedback messages
Exercise Specific Rules

• Designed individually for each exercise
• Checking special requirements for this exercise
  • Names taken from the exercise description
  • Structures specific for this exercise
• May handle tolerance for marginal flaws by user defined functions (e.g. for Levenshtein distance on strings)
  • Accept different spellings
  • Ignore whitespaces

<rule type="absence" points="10">
  <query>from x : V{Class}

    with stringLevenshteinDistance(x.name, "Pink Panther")<3
    report x.name as "name" end</query>

  <feedback>The diagram contains a class named "{name}" which seems to represent the phone company "Pink Panther". An entity with this name is supposed to be an instance and thus misplaced in a class diagram.</feedback>

</rule>
Generic Rules

- Independant from actual exercise
- Checking typical flaws
  - Missing directions, roles, cardinalities, ... on associations
- Checking style
  - Mixing of upper case and lower case letters
- Do not necessarily contribute to the grade, but only to the list of feedback messages

```
<rule type="absence" points="3">
  <query>
    from x : V{Association}, y : V{Property}
    with x --> y and isNull(y.lower) and isNull(y.upper)
    report x.name, y.name end</query>
  <feedback>The diagram contains associations without cardinalities.</feedback>
</rule>
```
Results

• Benefits
  • Good conformance between automated results and manual results
  • No complete sample solution needed
  • Flexible weighting of rules
  • Feedback messages for individual elements
  • Rules can be reused and shared

• Drawbacks
  • XMI 2 is a mess
  • Writing rules needs experience
  • General limitations in understanding natural language
Future Work

• More studies on conformance between automated results and manual results
• More diagram types
• Graphical feedback
• Other artefact types
  • Already used approach for static tests on Java syntax graphs with good results
• Answering your questions!