Generating and Evaluating Choices for Fixing Inconsistencies in UML Design Models

Alexander Egyed, Emmanuel Letier, and Anthony Finkelstein
UML Model
Video-on-demand (VOD) model in UML

Class diagram

Sequence diagram

Statechart diagram
of the class Streamer
### Consistency and inconsistencies Rules in UML

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td><code>operations=\text{message}.receiver\text{.base}.operations</code>&lt;br&gt;return <code>(operations-&gt;\text{name}-&gt;\text{contains}(message.name))</code>&lt;br&gt;Ensure that a message in the sequence diagram is declare as a method in the receiver’s class.</td>
</tr>
<tr>
<td>Inconsistency 1</td>
<td>message <code>\text{play}</code> is not defined as a method in Class <code>\text{Streamer}</code></td>
</tr>
<tr>
<td>C2</td>
<td><code>startingPoints = \text{find state transitions equal first message name startingPoints-&gt;exists(object sequence equal reachable sequence from startingPoint)}</code>&lt;br&gt;Ensure the behavior of a sequence of a message is allowed by state machine.</td>
</tr>
<tr>
<td>Inconsistency 2</td>
<td>Behavior of class <code>\text{Streamer}</code> (statechart) does not allow for message <code>\text{play}</code> to follow message <code>\text{connect}</code></td>
</tr>
<tr>
<td>C3</td>
<td><code>in=\text{message}.receiver.base.incomingAssociations;?\text{out=message.sender.base.outgoingAssociations;?\text{return \{in.intersectedWith(out)\}}}</code>&lt;br&gt;Ensure that the calling direction of the message is allowed by calling direction among classes.</td>
</tr>
<tr>
<td>Inconsistency 3</td>
<td>Calling direction of Message <code>\text{draw}</code> does not match association direction between classes <code>\text{Display}</code> and <code>\text{Streamer}</code></td>
</tr>
</tbody>
</table>
Inconsistencies Rules in UML from Meta-model
The Problem

• Identify all potential locations where to fix this inconsistency.
• Identifying how to change that location.

• Fixing Rules for all Locations
• Fixing rule for Consistency Rules
• Consistency rules differ among users
The Problem: Fixing Rules for all Locations

• Change method name
  • Message.Name

• Change the receiver of the message
  • Base.operation
The Problem: Fixing rule for Consistency Rules

Interplay among multiple consistency rule
The Problem: Consistency rules differ among users

- Different designers often use different consistency rules
- Resolution rule define for one designer is useless for another
The Approach:
How we generate the initial set of choices?

Choice Generation Functions:

• Generate all possible values that specific fields of a model element.
• Independent from the consistency rules

```
1  m:Message.receiver:
   choices = m.interaction.classifierRoles

2  m:Message.name:
   choices = {}
   foreach (method in m.receiver.base.methods)
     choices.insert(method.name)

3  ae:AssociationEnd.multiplicity
   choices = {1, 0..1, 1..n, 0..n}

4  c:Class.namespace
   choices = {}
   foreach (a in c.associations)
     foreach (oc in a.classifiers)
       choices.insert(oc.namespace)
```
What rules to re-evaluate to eliminate false choices?

- Choice generation function produce false rule.
- The elimination relay on the attributes.
- Check all the ways to fix the problem
  - Changing the receiver of the message
  - Changing the name of the message

Not Valid because of C2 (sequence of the message allowed)

Not Valid because of C1 (Method)
Impact of changes: Is the approach correct?

• Only correct if it can identify all consistency rule instances affected by the fix
• One issue affects multiple elements
• Limitation of work is it restricted to single change
The Approach

- 39 small to large UML models
- 24 types of consistency rules found in industry.
- 14 types of typical locations for fixing inconsistencies.
- Only 17% of all relevant locations were evaluated
The Approach

• False location is a location for which no valid choice exists, bar suggest 11.2%

• By exploring choices we are able to automatable detect these false locations

Figure 10. Number of Locations for Fixing Inconsistencies (with/without false ones)
Conclusion

Strength of the approach

• Approach does not suggest false choices
  • Able to identify all consistency rules affected by the rules automatically
• Use a white-box constraints.
• The approach was made on UML, however it can be transfer to other meta-models.
• The approach tool provide “On-line” suggestions of choices.
• 40% had a single valid choice
  • Possible automated correction
Weaknesses of the approach

• Technique can make a change in one location at a time
• The approached was checked on IBM Tool only.
• The approach tested on small group of models.
Discussion

• Do you think in the future there will be possible auto-correction?
• What rollback mechanism shall be put? If any?
• Do you think that we need to have logs in the system?
• Do you think the approach be develop from the code to the model?
  (Like MVVM)