Abstract

Aspect-Oriented Programming is a software engineering paradigm that offers new constructs, such as join points, pointcuts, advices, and aspects in order to improve separation of crosscutting concerns. The new constructs bring new types of programming faults with respect to crosscutting concerns, such as incorrect pointcuts, advice, or aspect precedence. In fact,
existing object-oriented testing techniques are not adequate for testing aspect-oriented programs. As a result, new testing techniques must be developed. In this paper, an approach based upon UML activity diagram for testing aspect-oriented programs is presented. The proposed approach focuses on integration of one or several crosscutting concerns to a primary concern and tests whether or not an aspect-oriented program conforms to its expected crosscutting behaviors. The proposed approach generates test sequences based on interaction between aspects and primary models, and verifies the execution of the selected sequences. It also, follows an iterative process which causes to discover faults easily and quickly. The approach is based on several test criteria that we defined. To illustrate the approach, we use a case study which its results show that the approach is capable of revealing several aspect-specific faults.

Reference

Testing Aspect-Oriented Programs with UML Activity Diagrams


Index Terms

Computer Science
Software Engineering
### Key words

<table>
<thead>
<tr>
<th>Aspect-Oriented Programming</th>
<th>Model-Based Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Aspect-Oriented Modeling</td>
</tr>
<tr>
<td></td>
<td>UML Activity Diagrams</td>
</tr>
</tbody>
</table>

Test Sequences