Inconsistencies within UML models or between UML models and requirements, standards, or other design artifacts may result directly in software defects. IV&V capability for systematic UML model consistency checking can help prevent these defects. Audit and evaluation of inconsistencies may also provide clues to other deficiencies in model correctness or completeness. This poster was developed through a NASA IV&V Summer College Internship Program (SCIP) project.

Classes of Inconsistency

- **Inter-Model Inconsistency**: Between models or diagrams at different levels of abstraction.
- **Intra-Model Inconsistency**: Within a Model or Diagram.
- **Static Inconsistency**: In Model Structure, Content or Syntax.
  - Are we building the system right?
- **Dynamic Inconsistency**: Between modeled and expected behavior.
  - Are we building the right system?
- **Semantic Inconsistency**: Different Meanings.

Examples of Inconsistency

- Horizontal Intra-Model Inconsistencies
  - Between models or diagrams at the same level of abstraction.
- Vertical Inconsistency
  - Between models or diagrams at different levels of abstraction.

Approaches to Finding Inconsistency

- UMLAnalyzer is an incremental consistency checking tool. The tool locates inconsistencies and presents them in a graphical interface. The interface allows the user to select a specific inconsistency, choose how to fix it, and reanalyze the model only if the fix made anything else inconsistent. This process is repeated until all defects are resolved. UMLAnalyzer includes both UML and OCL model checking capability.

UML/Analyzer is an incremental consistency checking tool. The tool locates inconsistencies and presents them in a graphical interface. The interface allows the user to select a specific inconsistency, choose how to fix it, and reanalyze the model only if the fix made anything else inconsistent. This process is repeated until all defects are resolved. UMLAnalyzer includes both UML and OCL model checking capability.

Approaches to Finding Inconsistency

- **OCLE**: Provides an environment for formulating OCL rules and for detecting static and dynamic inconsistency at the model level of system abstraction. In addition to automating checks against UML Well-Formedness rules, OCLE automates Methodological Rules, Profile Rules or Target Implementation Language Rules expressed in OCL. For example, the OCLE tool can be used to check that object diagrams conform to a class model, i.e., that invariants specified in the class model hold in the object diagrams. Dynamic consistency checking is supported via translation of OCL extended UML to Java source code.

Approaches to Finding Inconsistency

- **SPIN**: Simple PROMELA Interpreter. It is a prominent tool in the UML model checking literature. SPIN reports deadlocks, unspecified receptions, incompatibilities, race conditions, and unwarranted assumptions about the relative speeds of processes. SPIN uses PROMELA (Process Meta Language) to check models. However, SPIN is not made specifically for UML. Use of SPIN requires translation from UML to a SPIN model. XSPIN could potentially be used to integrate PROMELA and SPIN with Together or other Eclipse-based UML modeling tools for UML model consistency checking.

Table 2: Approaches for Dealing with UML Inconsistencies

<table>
<thead>
<tr>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meta-Modeling</td>
<td>Natural extension to the source code</td>
<td>Strict commitment to the chosen language</td>
</tr>
<tr>
<td>Constraint Language</td>
<td>Enhanced meta-language allowing for better constraints</td>
<td>Non-trivial implementation and usually needs access to some unavailable meta-model data</td>
</tr>
<tr>
<td>Formal Notations</td>
<td>Ease of check consistency and availability of consistency management frameworks</td>
<td>Could be inefficient if not designed to implement and difficult to integrate with tools</td>
</tr>
</tbody>
</table>

Principal Author:
Chance B. Cover
NASA IV&V Summer College Internship Program (SCIP)

NASA POC & Presenter:
Michael Facemire
L-3 Communications
Michael.L.Facemire@ivv.nasa.gov