UML Model Consistency Checking

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. It presents classes of UML model inconsistency, examples of UML model inconsistency, approaches to identifying inconsistency, and some consistency checking tools of interest.

Classes of Inconsistency

Vertical Inconsistency
Between models or diagrams at different levels of abstraction

Horizontal Inconsistency
Between models or diagrams at the same level of abstraction

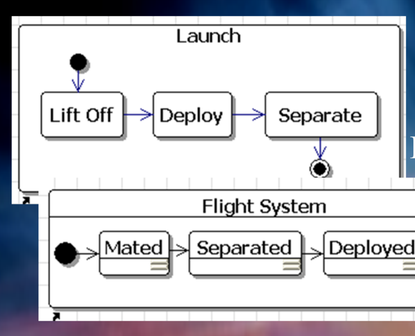
Inter-Model Inconsistency
Between different models

Intra-Model Inconsistency
Within a Model or Diagram

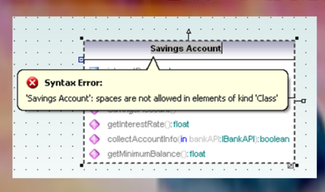
Static Inconsistency
- In Model Structure, Content or Syntax
- Are we building the system right?

Dynamic Inconsistency
- Between modeled and expected behavior
- Are we building the right system?

Semantic Inconsistency
Different Meanings



Syntactic Inconsistency
Between Model and Language Rules



Horizontal Intra-Model Inconsistencies

Source: Management and verification of the consistency among UML Models, Atsushi Ohnishi (Department of Computer Science, Ritsumeikan University, Shiga 525-8577, Japan)

	Class diagram(C)	State chart(S)	Sequence chart(Q)	Collaboration diagram(L)	Use case diagram(U)
State chart (S)	1) Class of (S) and classes in (C) 2) confirmation of classes without state charts. 3) Attributes defining states in (S) and attributes in (C) 4) Range of attribute values in (C) and (S) 5) actions, activities in (S) and methods in (C)				
Sequence chart(Q)	1) Objects in (Q) and classes in (C) 2) Messages between objects in (Q) and associations between coressponding classes in (C) 3) Messages in (Q) and methods in (C)	1) Object in (Q) and class of (S) 2) Messages in (Q) and actions, activities in (S)			
Collaboration diagram(L)	1) Objects in (L) and classes in (C) 2) Messages between objects in (L) and associations between corresponding classes in (C) 3) Messages in (L) and method in (C)	1) Objects in (L) and class of (S) 2) Messages in (L) and actions, activities in (S)	1) Objects in (L) and in (Q) 2) Messages in (L) and in (Q) for directions, sequence, source, and destination		
Use case diagram(U)	1) Actors in (U) and classes in (C) 2) Use cases in (U) and methods in (C)	1) Actors in (U) and class of (S) 2) Use cases in (U) and actions, activities in (S)	1) Actors in (U) and objects in (Q) 2)use cases in (U) and messages in (Q)	1) Actors in (U) and objects in (L) 2) Use cases in (U) and messages in (L)	
Activity diagram(A)	1) Classes in (A) and in (C) 2) Actions in (A) and methods in (C) 3) Control flows between classes in (A) and associations in (C)	1) Classes in (A) and class of (S) 2) Actions in (A) and actions, activities in (S)	1) Classes in (A) and objects in (Q) 2) Actions in (A) and messages in (Q) 3) Control flows between classes in (A) and messages in (Q)	1) Classes in (A) and objects in (Q) 2) Actions in (A) and messages in (L) 3) Control flows between classes in (A) and message in (L)	1) Classes in (A) and actores in (U) 2) Actions in (A) and use cases in (U)

Figure 1 Knowledge-based verification items among UML models

Examples of Inconsistency

Approach	Advantages	Disadvantages
Meta-Modeling	Natural extension to the language	Strict commitment to the chosen semantics
Constraint Language	Enhanced meta-language allowing for better constraints	Non-trivial implementation and usually needs access to some unavailable meta-model data
Formal Notations	Ease of check consistency and availability ofconsistency management frameworks	Could be inefficient (not scalable) to implement and difficult to integrate with tools

Table 2: Approached for Dealing with UML Inconsistency

Source: An Overview of UML Consistency Management. M. Elaasar, L. Briand

Approaches to Finding Inconsistency

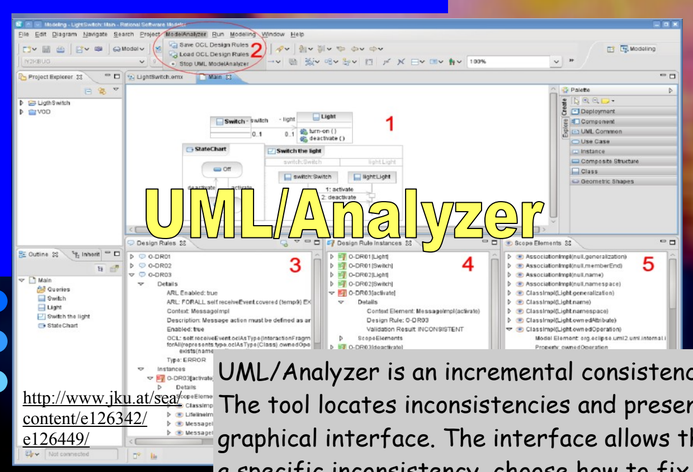
Model Transforms

Model Audits

Animation/Simulation

UML Model Checking Tools

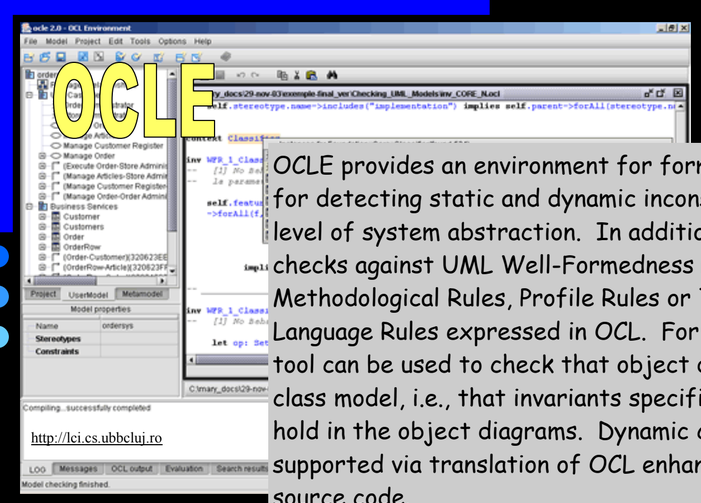
UML/Analyzer



http://www.jku.at/seal/content/e126342/e126449/

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 re-analyze the model see if the fix made anything else inconsistent. This process is repeated until all defects are resolved. UML/Analyzer includes both UML and OCL model checking capability.

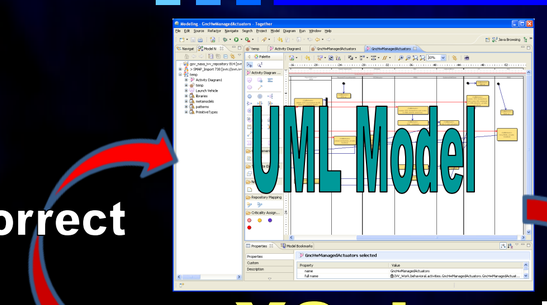
OCLE

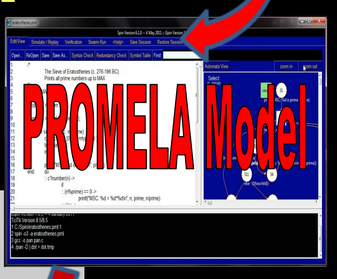


http://lci.cs.ubbcluj.ro

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 enhanced UML to Java source code.

XSpin





http://spinroot.com/

SPIN (Simple PROMELA INterpreter) is a prominent tool in the UML model checking literature. SPIN reports deadlocks, unspecified receptions, incompleteness, 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.

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

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

NASA Independent Verification and Validation Facility
Fairmont, West Virginia