

## Representations, Methods and Tools

### 1. Representations

Effective verification, including informal testing, requires that the system to be verified is specified in an unambiguous representation. This means either formal languages (Z, ACL2, SMV or Promela, etc.) modeling languages (xUML, etc.) or subsets of standard programming languages (Java, etc.) or specialized languages such as Haskell or logic programming languages. We will focus on modeling languages such as xUML which can be translated to either conventional procedural languages or formal languages and on subsets of Java which have readily formalizable semantics for which there exist verification tools.

### 2. Tools

The tools to be used implement static analysis, automated test generation, model checking and theorem proving. The tools to be used include static analysis systems (ESC/Java, etc.), static analysis based test generation tools such as are in Eclipse, model checkers (SPIN, NuSMV, COSPAN, etc.), complete model checking systems such as Bandera and Objectcheck, etc. and theorem provers such as ACL2, PVS, Maude, etc.