



Test Case Generation For Programming Language Metamodels

Hao Wu haowu@cs.nuim.ie

Computer Science Department,
National University of Ireland, Maynooth.

<http://www.cs.nuim.ie/research/pop/>

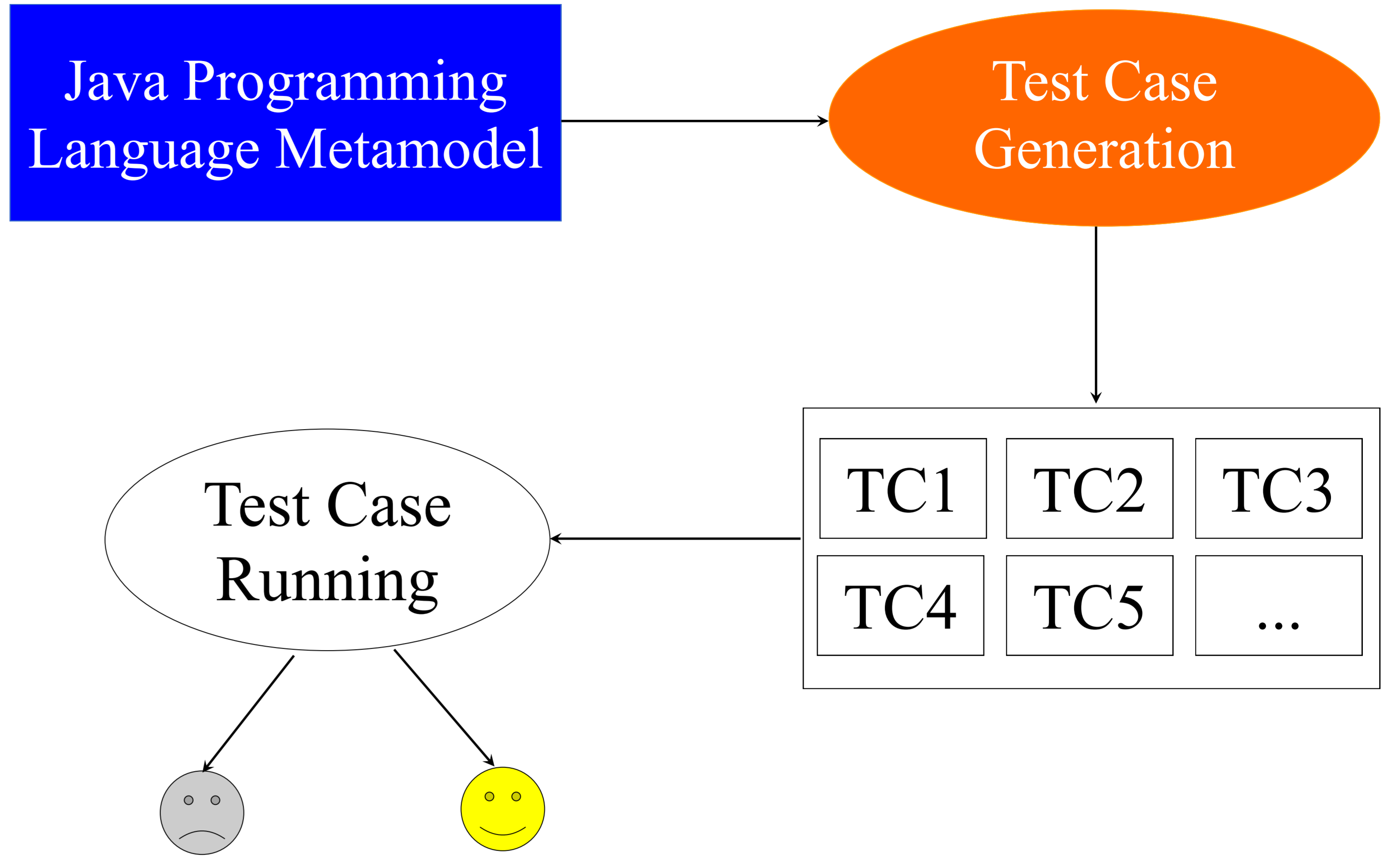
NUI MAYNOOTH
Ollscoil na hÉireann Má Nuad

Challenge:

Given a programming language metamodel, how can we generate an *appropriate* test suite to show that it is valid?

Motivation:

1. Examining the automatically-generated test cases would help to develop the modeler's understanding of the metamodel, and help to increase confidence in its validity.
2. The generated test cases can be used as test inputs for tools that process the language.



What do we mean by an appropriate set of test cases?

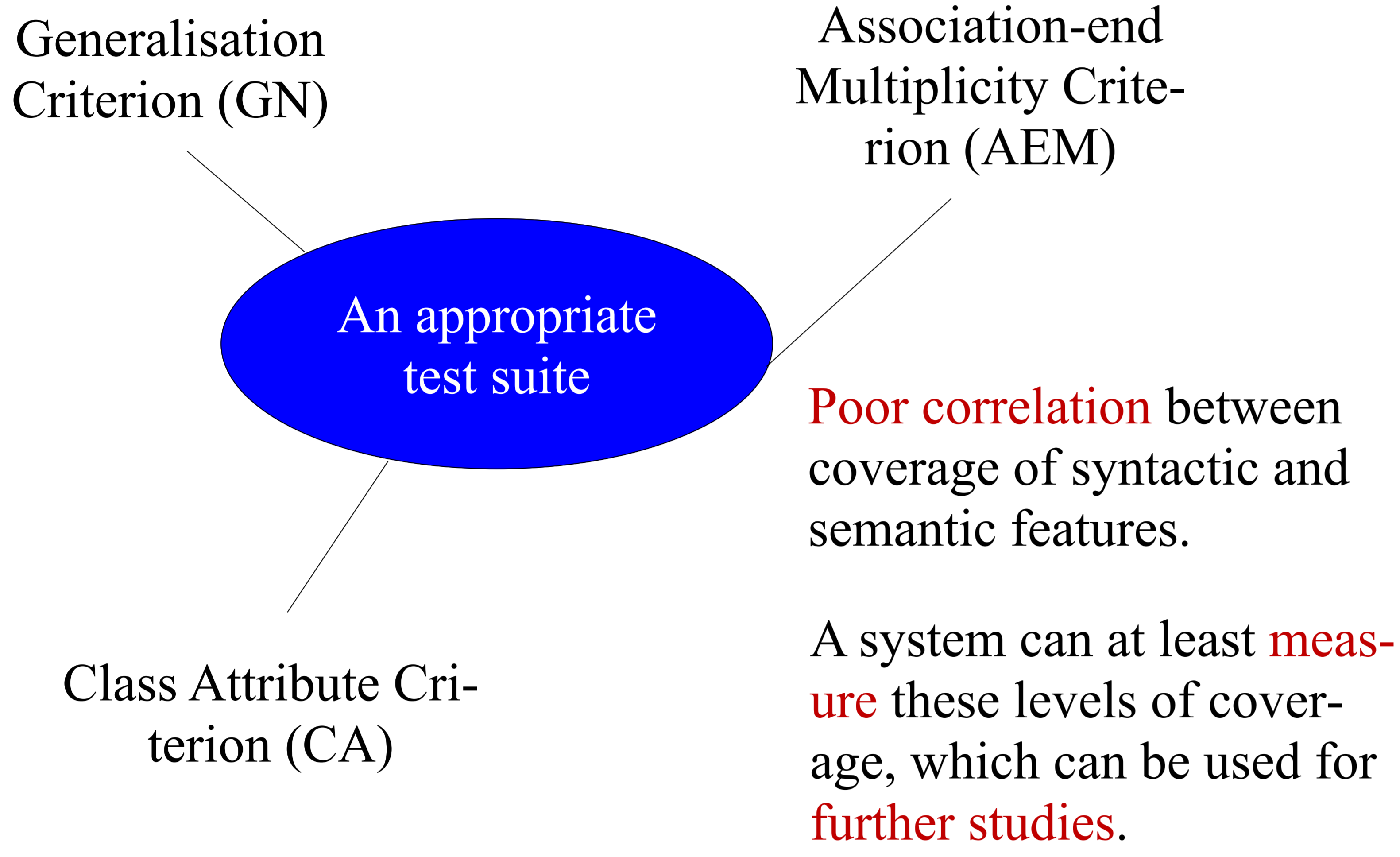
Use coverage criteria to tell the adequacy of a set of test cases. More specifically, coverage criteria for UML class diagrams.

Three Coverage Criteria [1]:

Generalisation coverage: how to measure inheritance relationships.

Association-end multiplicity coverage: measures association relationship between classes.

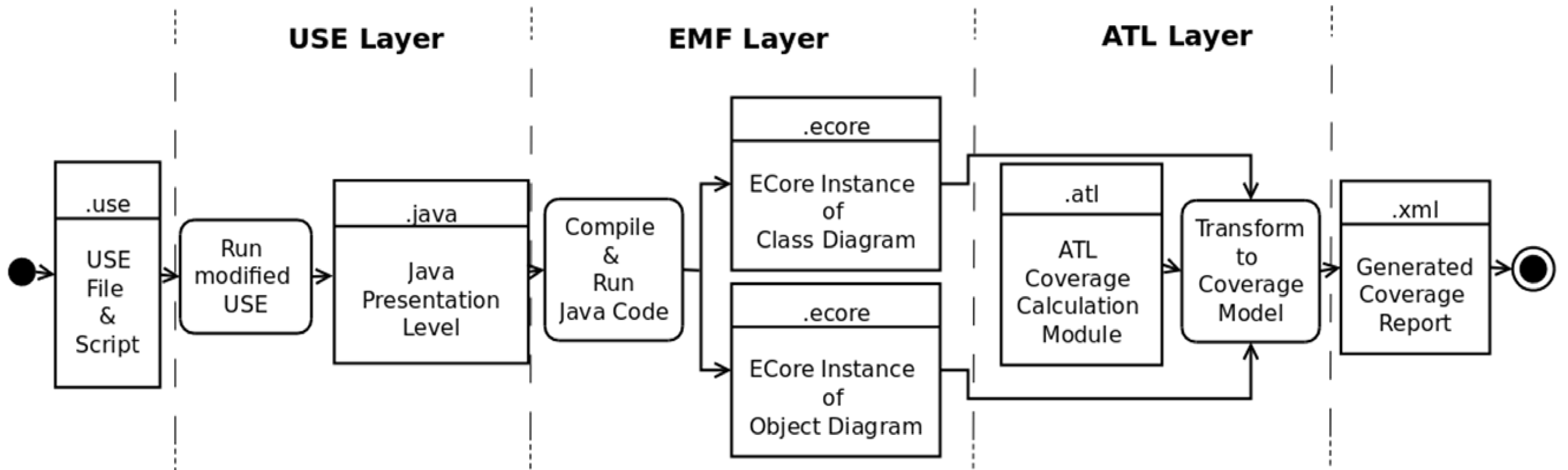
Class attribute coverage: measures the set of representative attribute value combinations in each instance of class.



1. Andrews, A., France, R., Ghosh, S., Craig, G.: Test adequacy criteria for UML design models. *Soft. Test. Verif. & Reliability* 13(2) (April/June 2003) 95-127

Our Current Work

In our work to date we have constructed a tool-chain which, when given a UML class diagram and a set of UML object diagrams, will calculate the three coverage measures [2].



This tool chain uses the USE tool as a parser and validator for the class and object diagrams, and uses the Eclipse Modeling Framework (EMF) to represent these as instances of the UML metamodel [3].

We use a part of Software Metrics Metamodel (SMM) as our coverage metamodel for storing the results of coverage measures.

Finally, we have written a transformation to calculate the coverage measures using ATL (ATLAS Transformation Language).

2. Wu, H., Monahan, R., Power, J.F.: Using ATL in a tool-chain to calculate coverage data for UML class diagrams. In: 2nd International Workshop on Model Transformation with ATL, Malaga, Spain (June 2010)

3. Gogolla, M., Büttner, F., Richters, M.: USE: A UML-based specification environment for validating UML and OCL. *Sci. Comp. Prog.* 69(1-3) (2007) 27-34

Future Work

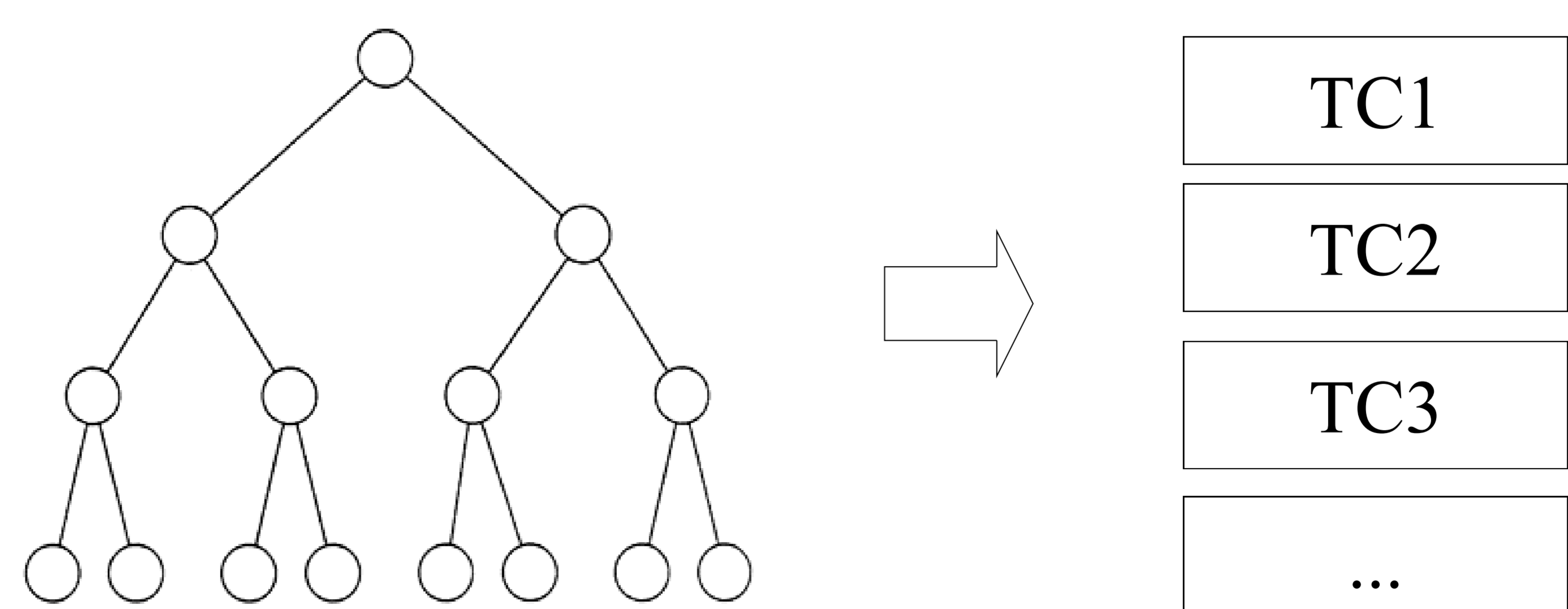
We intend to extend coverage measures for class diagrams to deal with the associated Object Constraint Language (OCL) constraints.

Next Phase:

Studying approaches to test case generation possibly exploiting third-party SAT solvers to control model generation.

Final Phase:

We intend to use OCL-based queries across the metamodel to specify the kind of model to be generated.



OCL Query Example:

```
context While_Statement inv:
  self.nested_level >= 1
```

To specify the test case that have at least one nested loop statement.

This work is supported by a John & Pat Hume Scholarship from NUI Maynooth.