

Towards the re-usability of software metric definitions at the meta level

Jacqueline A. McQuillan

Research Advisor: James F. Power



Principles of Programming Research Group

Department of Computer Science

National University of Ireland Maynooth, Co. Kildare, Ireland

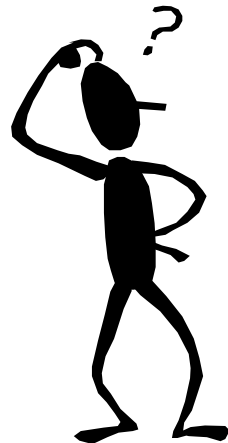


Overview

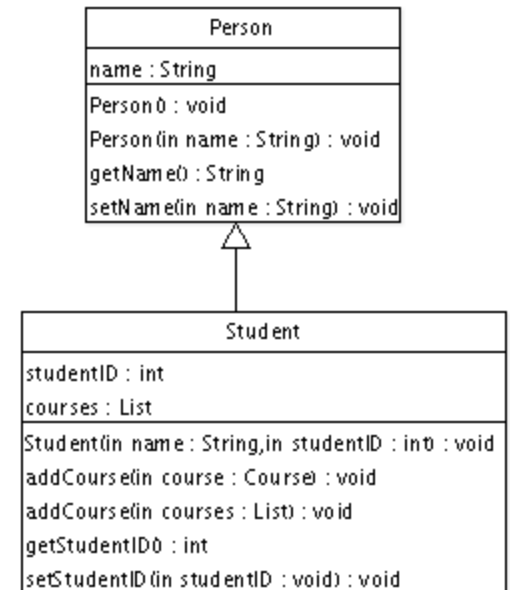
- Motivation
- Metamodels and the Object Constraint Language (OCL)
- Approach to defining software metrics
- Applying the approach to the Unified Modeling Language (UML)
- The dMML tool
- Summary
- Discussion

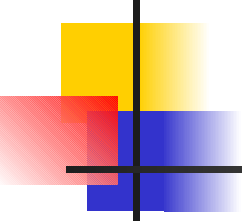
1 Motivation

- Software metrics
- No standard terminology or formalism
- Imprecise and ambiguous definitions
- Different interpretations



9
5
3
4
Number of Methods?



- 
-
- Difficult to conduct repeatable experiments
 - Cannot compare results obtained by different researchers
 - Hinders empirical validation
 - Metrics need to be empirically validated in order to be widely accepted



2 Metamodels and the OCL

- The Object Constraint Language (OCL) 2.0 and language meta-models as a mechanism for defining metrics

metamodel?

OCL?



2.1 Metamodels

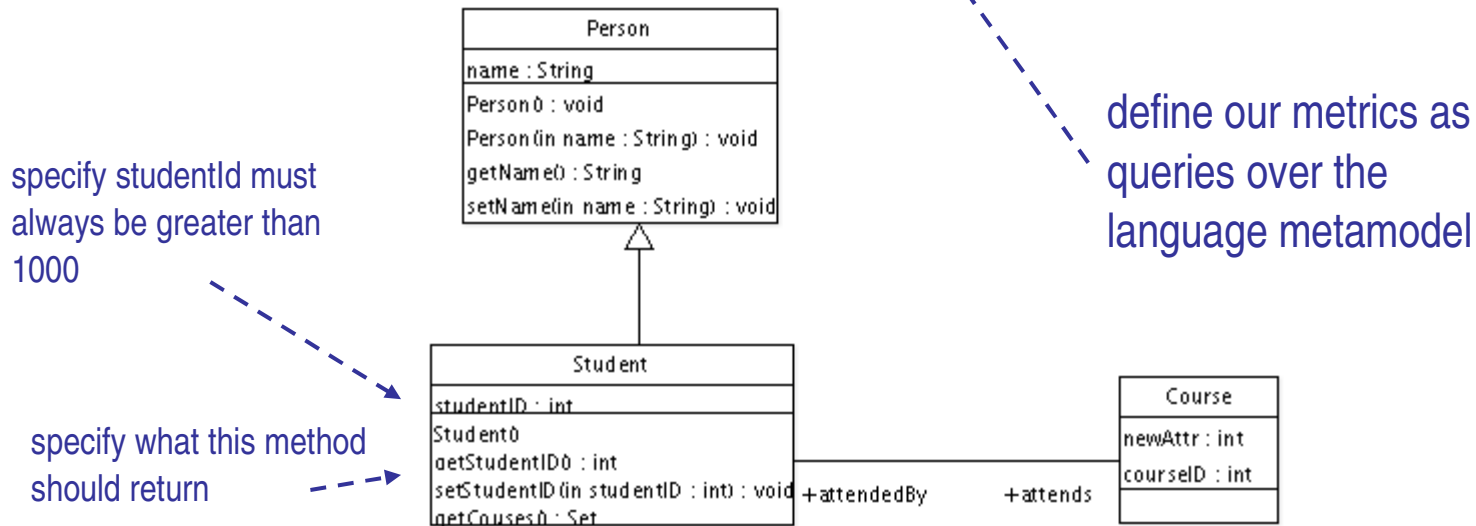
- What is a meta-model?
 - a model
 - used to define a language
 - specifies what constructs are allowed in the language and the relationships between the constructs eg. UML, Java metamodels



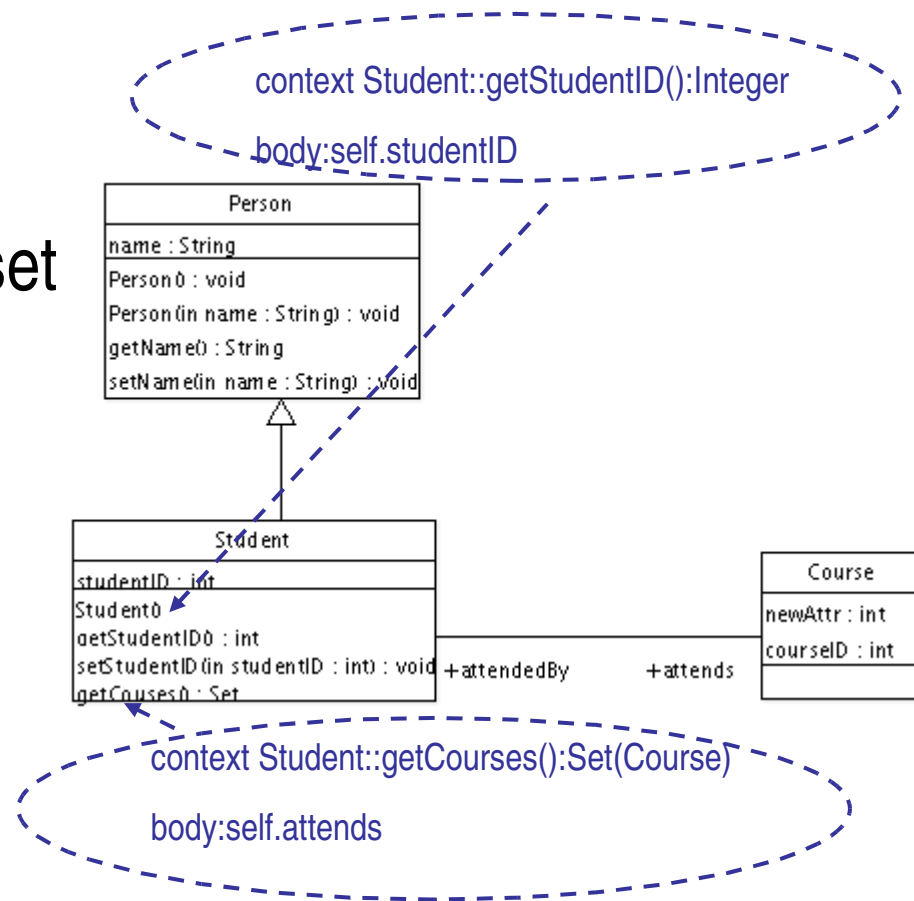
2.2 The Object Constraint Language

- Part of the Unified Modeling Language (UML)
- UML
 - Standard Modelling language
 - Used to develop models of software systems
 - Sometimes cannot capture all the details of a system that you require with UML diagrams
 - Offers a textual language (OCL)

- Standard language that allows you to write constraints and queries over object oriented models in a clear and unambiguous manner



- This example states that the operation *getCourse* will always result in the set of Courses associated with the Student





3.1 Approach to defining metrics

- Express metrics as OCL queries
- Over a language metamodel



Approach by Baroni et al.

- Baroni et al. propose expressing design metrics as OCL queries over the UML 1.3 metamodel
- This approach involves modifying the metamodel by creating the metrics as additional operations in the metamodel and expressing them as OCL conditions

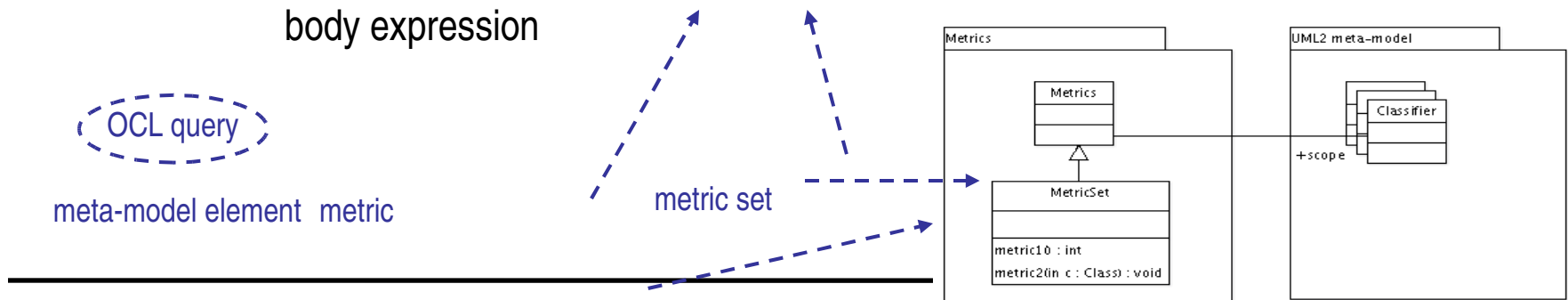


Extensions to the approach

- Decoupling the metric definitions from the metamodel
- Express them as body expressions
- Made approach generalizable by parameterising by the metamodel and metric definitions
- Implementing the approach based definitions on UML2.0

3.2 Implementing the approach

- Create a metrics package to the meta level
- Defining a new metrics set is a three step process:
 - A class is created in the metrics package corresponding to the metric set; any auxiliary operations are defined in this class.
 - For each metric, an operation in the class is declared, parameterised by the appropriate elements from the metamodel.
 - The metrics are defined by expressing them as OCL queries using the OCL body expression



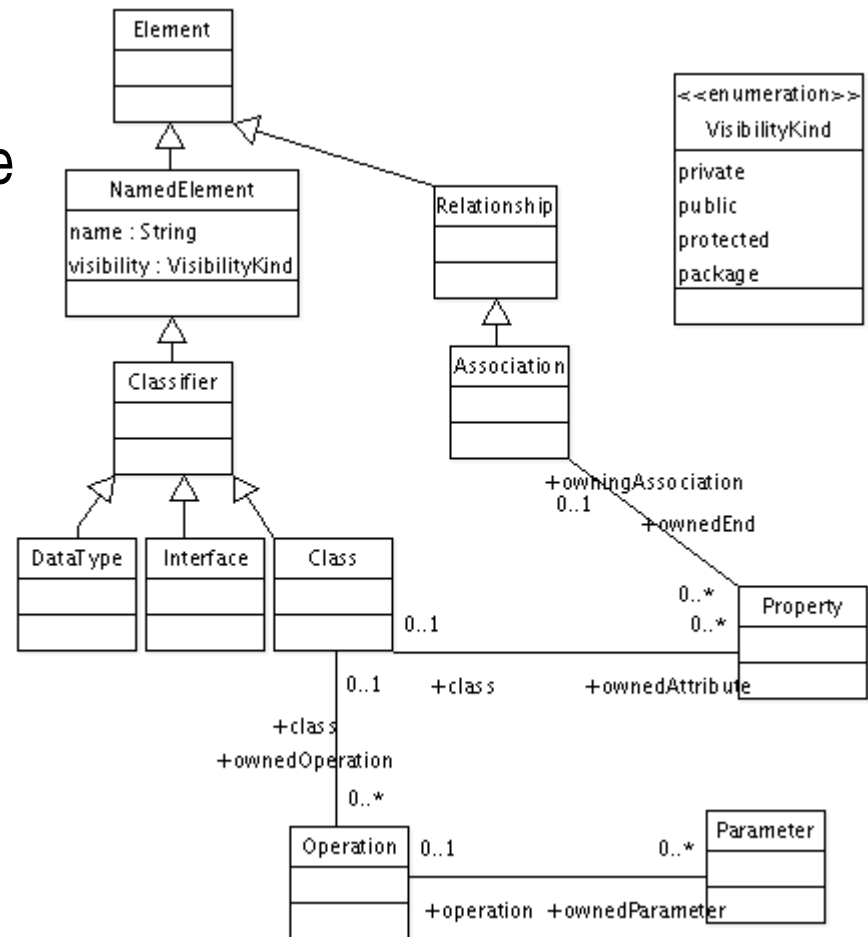


4 Applying the approach

- Chose UML-based metrics
 - Class diagrams
- Define software metrics using the Object Constraint Language (OCL) 2.0 and the Unified Modeling Language (UML) 2.0 meta-model

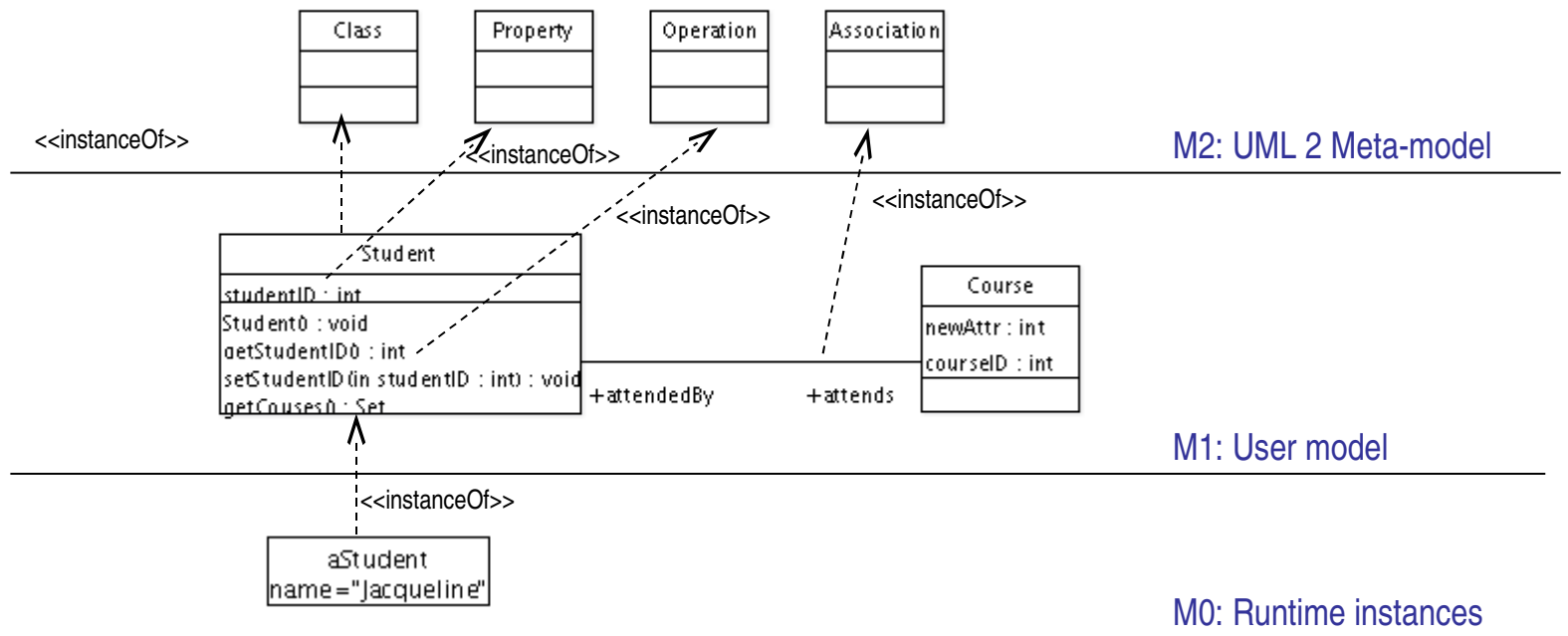
4.1 The UML metamodel

- The ***UML metamodel*** is a model that is used to define the ***UML***
- Specifies the constructs that may be used in a UML model and the relationships between these constructs



Part of the UML metamodel (simplified)

4.2 Metamodel hierarchy

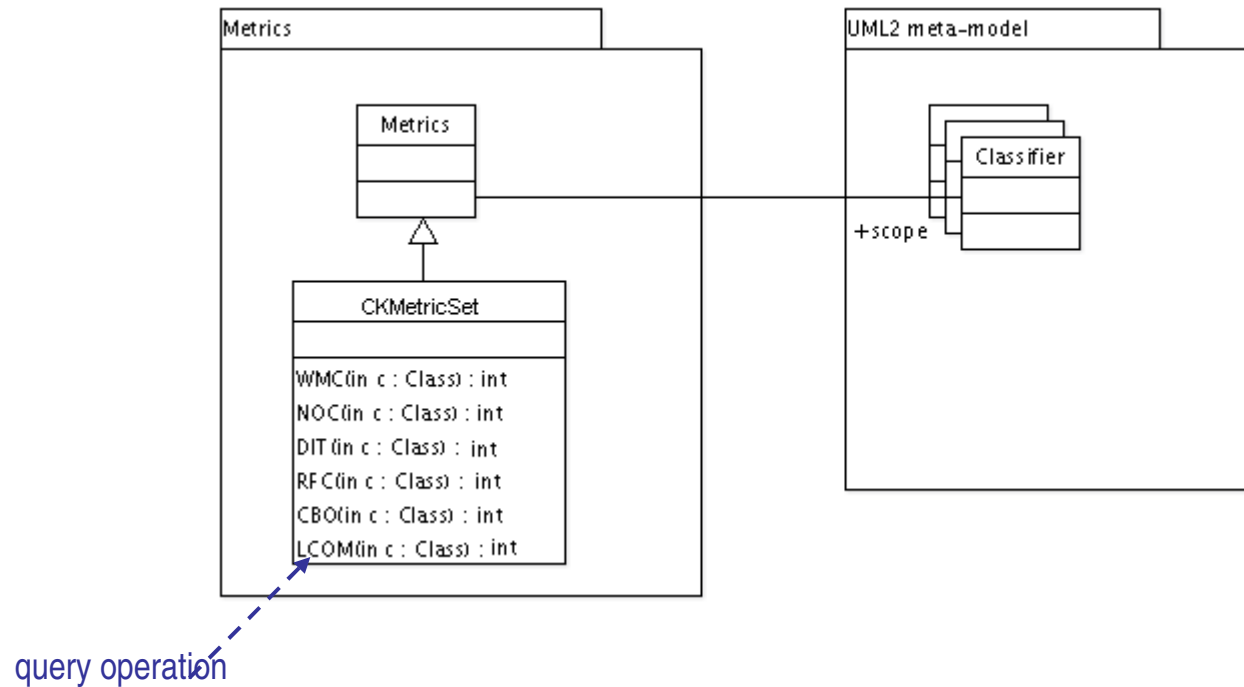




4.3 C&K metrics suite

- Chidamber & Kemerer Metric Suite
 - Weighted Methods per Class (WMC)
 - Number of Children (NOC)
 - Depth of Inheritance Tree (DIT)
 - Response for a Class (RFC)
 - Coupling Between Objects (CBO)
 - Lack of Cohesion of Methods (LCOM)

CKMetrics model





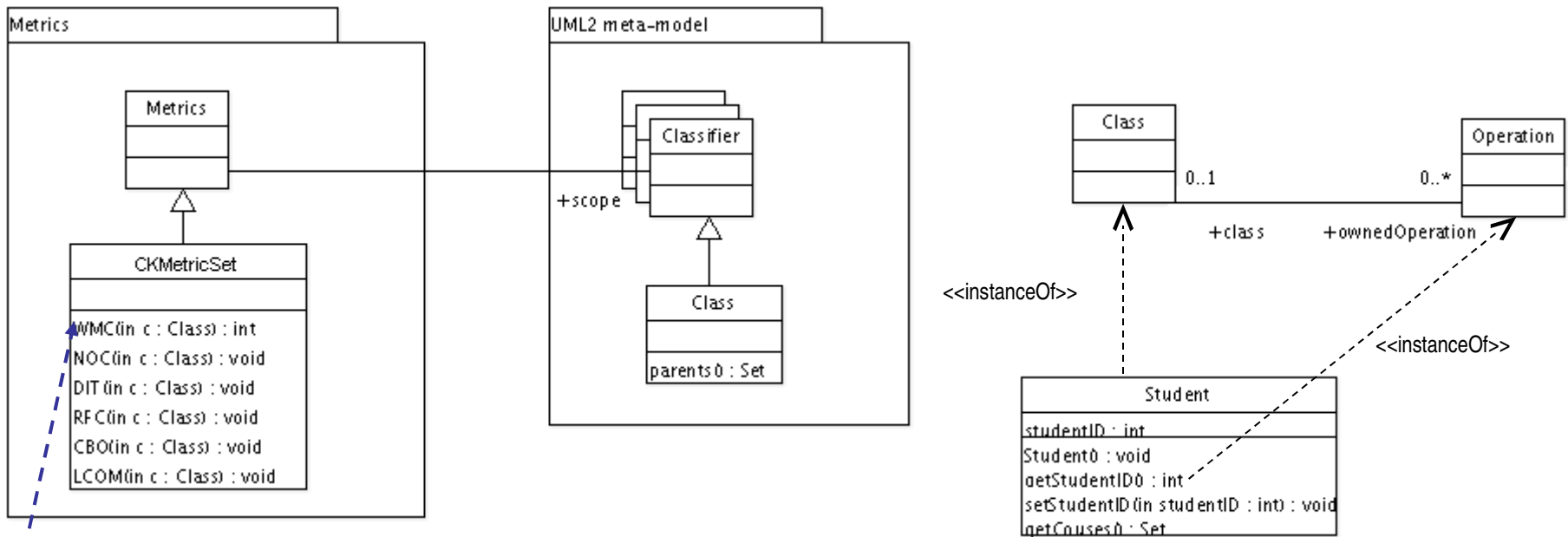
WMC Metric

- Consider a Class C , with methods M_1, \dots, M_n . Let c_1, \dots, c_n be the static complexity of the methods

$$WMC = \sum_{i=1}^n c_i$$

- If all static complexities are considered to be unity, $WMC = n$, the number of methods

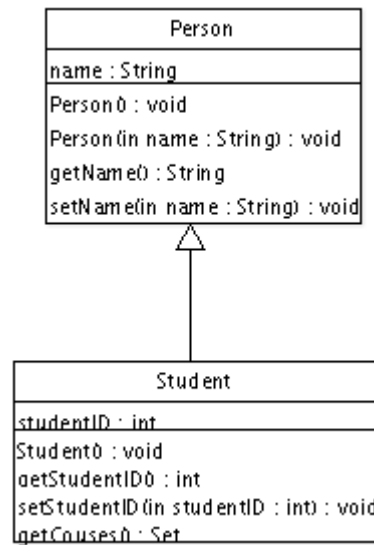
OCL Definition



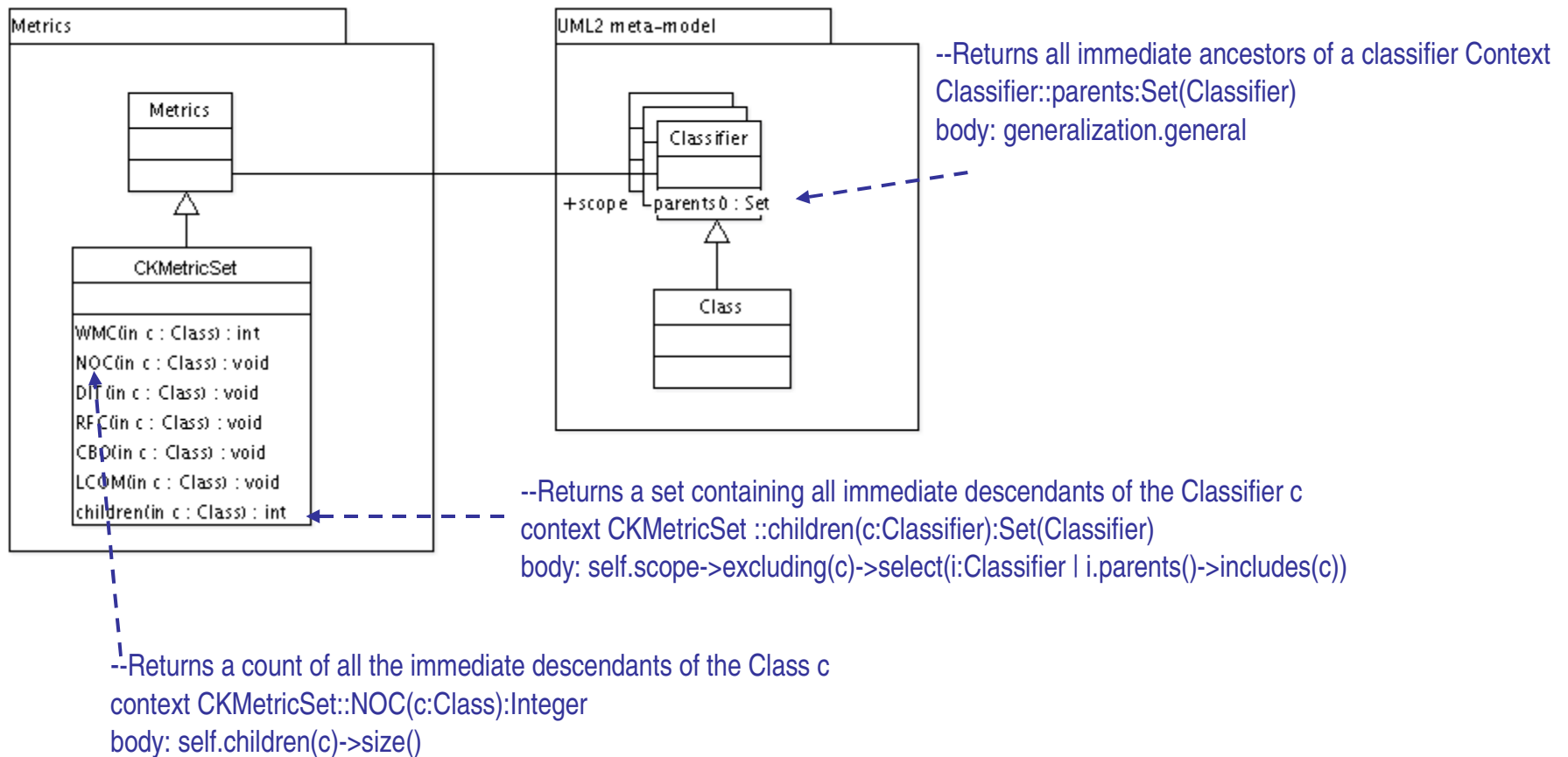
--Returns a count of all the operations belonging to the Class c context
 CKMetricSet::WMC(c:Class):Integer
 body: c.ownedOperation->size()

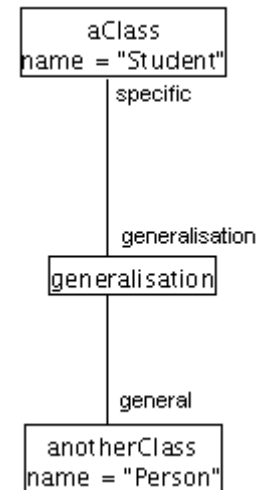
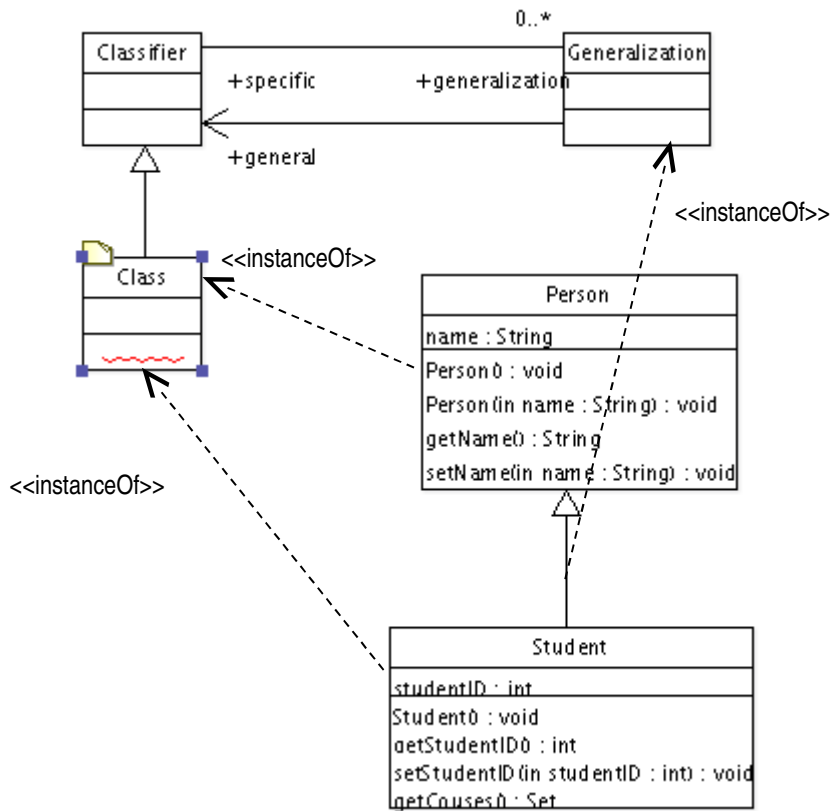
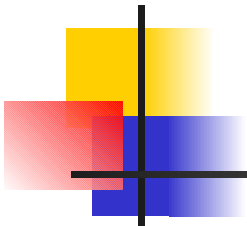
NOC Metric

- NOC = number of immediate sub-classes subordinated to a class in the class hierarchy



OCLE Definition







5 dMML

- Plug-in for Eclipse
 - An extensible, open source IDE (integrated development environment)
- Dependent on *Octopus*
 - Checks syntax of OCL expressions
 - Transforms UML models , including OCL expressions, into Java code



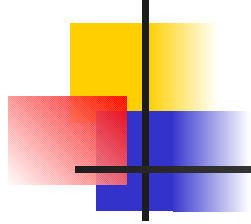
6 Discussion

- Have we completely removed the problem of ambiguity or has this just moved from the definitions to the underlying meta-models as ultimately the definitions rely on the meta-model
Example of number of methods
- How do we know that the definitions are correct?
And ultimately that the metric results we obtain are correct for the software system



7 Summary

- No standard terminology or formalism
- Presented an approach to define software metrics
- Applied approach to UML-based metrics using the UML 2.0 metamodel
- Defined C&K metrics
- Developed a prototype tool that implements the approach
- Identified some questions/limitations



Thank You!