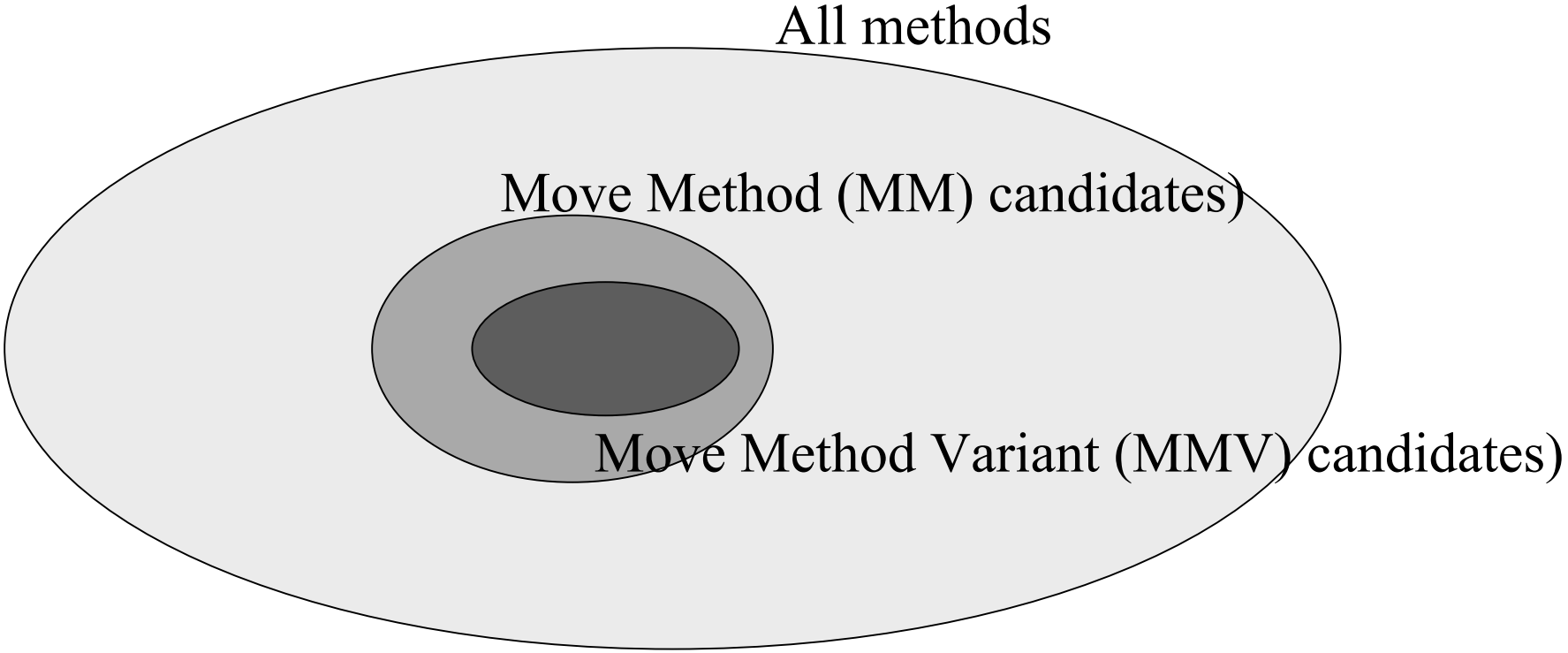
Qualitative Preconditions for Refactoring

Analysis procedures

Bart Du Bois

FLOWER - 23/01/2006

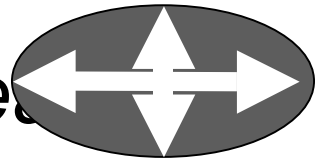
Estimating the effect of a refactoring



Precondition(MMV) => Precondition(MM)
Postcondition(MMV) => Postcondition(MM)

e.g. Postcondition(MMV) = Postcondition(MM) + less coupling

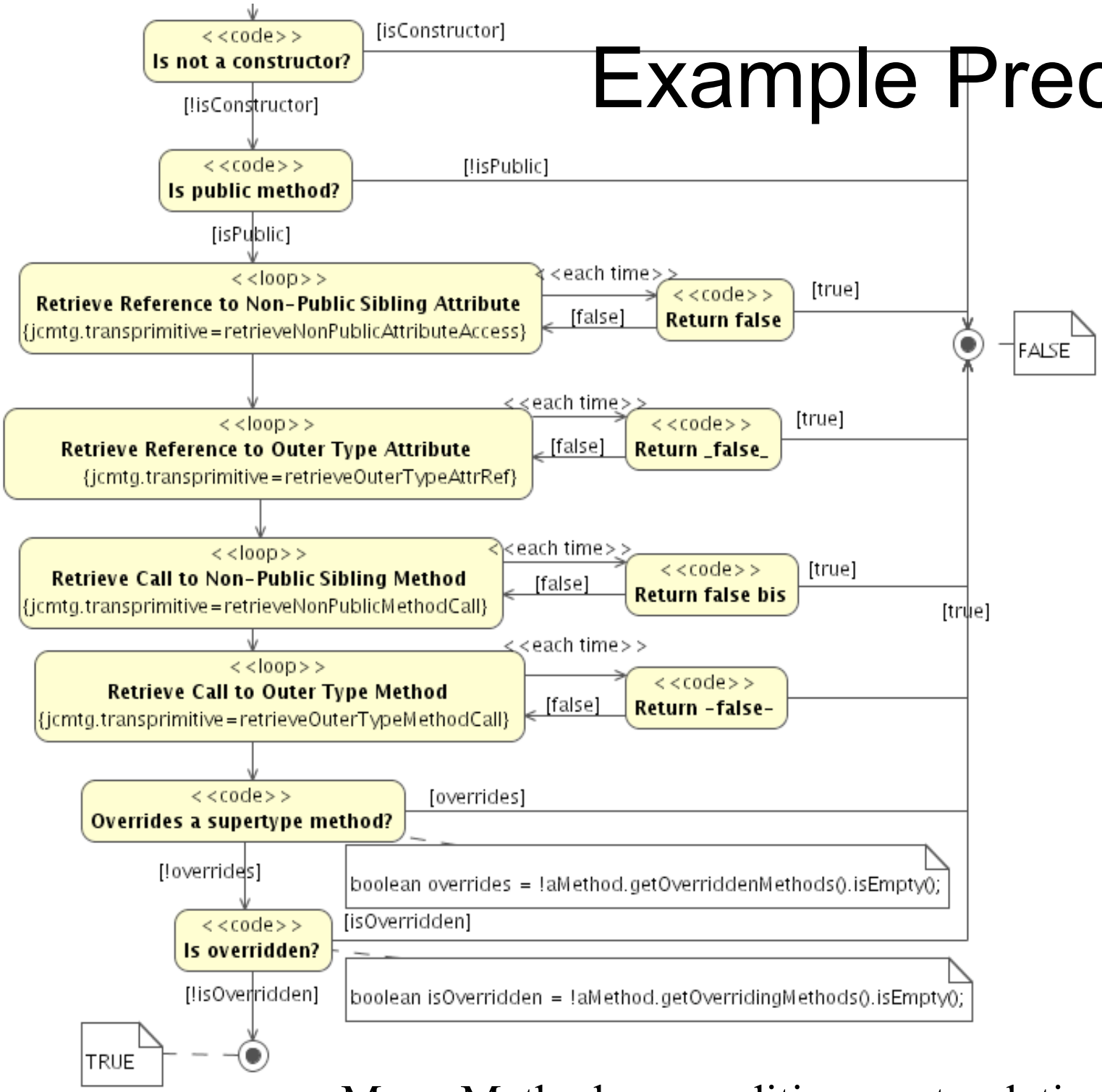
Validating Refactoring Variants



- Evaluate Preconditions to assess *applicea*
- Calculate Metrics to validate *effect estimation*
  - *Postcondition(MMV) \ Postcondition(MM)*

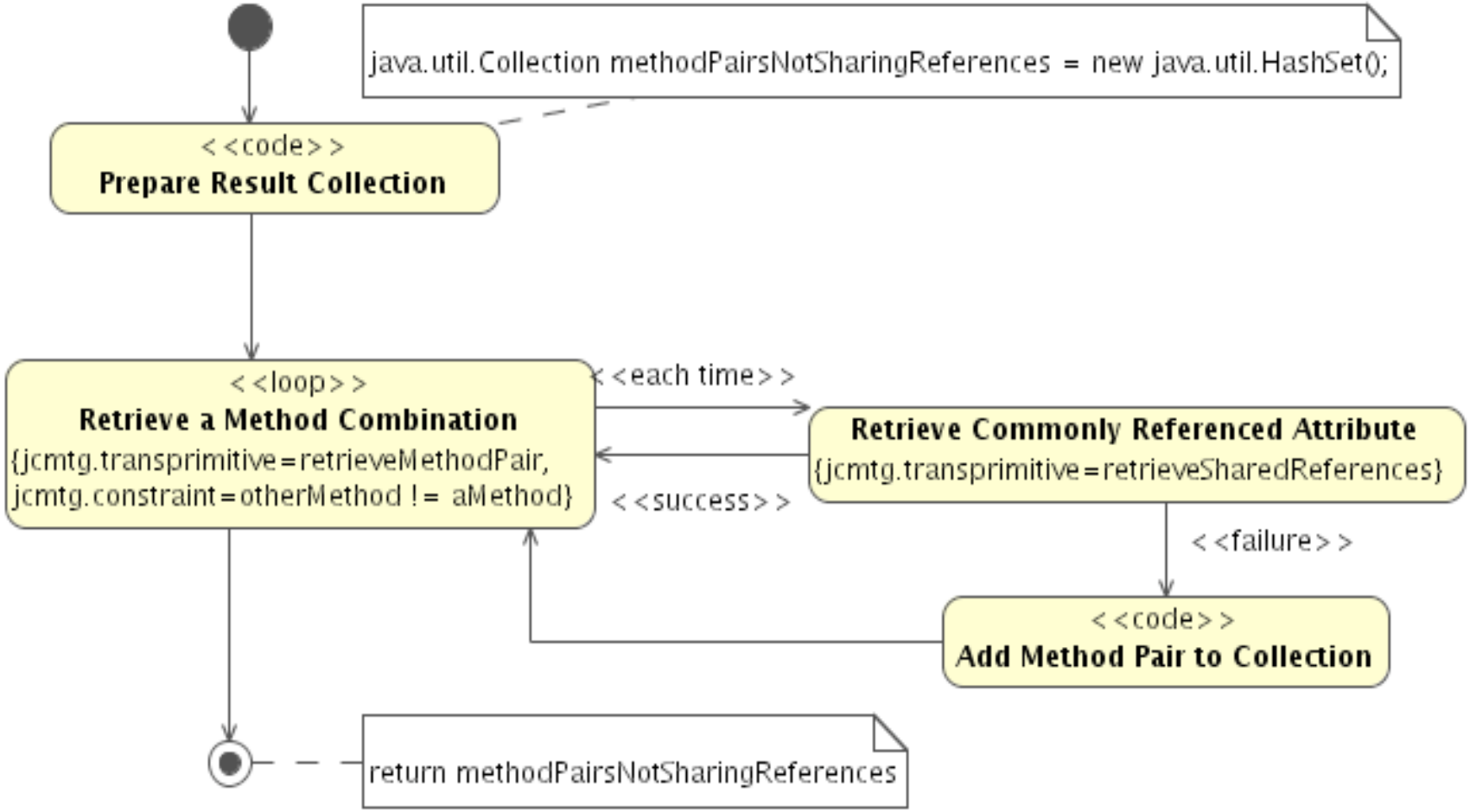
- Requires instances of a GRAMMY/FAMIX-alike metamodel
  - *Nodes*: Packages, Types, Methods, Attributes, Parameters
  - *Edges*: Composition, Inheritance, Typing, Invocation, Reference

Example Precondition



Move Method precondition w.r.t. relationship between m_{src} and

Example Metric



Lack of Cohesion of Methods (LCOM1)

JDT2MDR and MotMot

A succesful collaboration

People involved

LORE: JDT2MDR

- Bart DB
  - Transformation framework
- Bart VR
  - Extensions for testing data

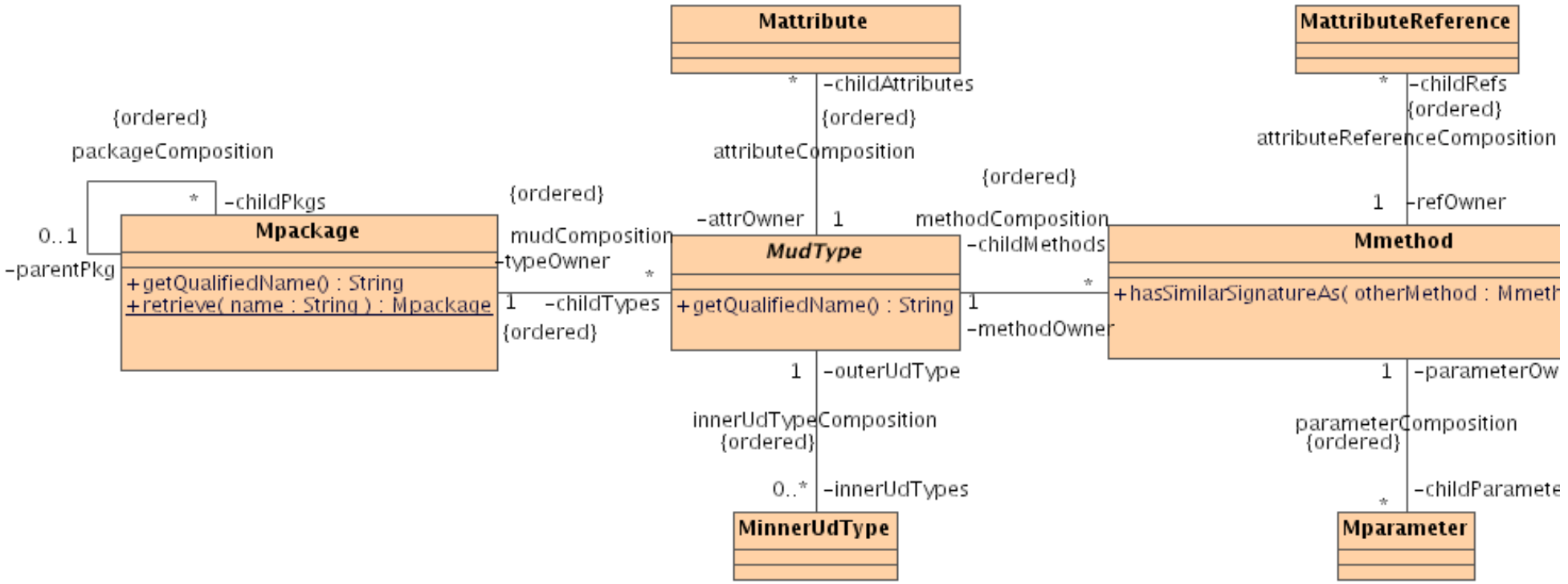
FOTS: MotMot

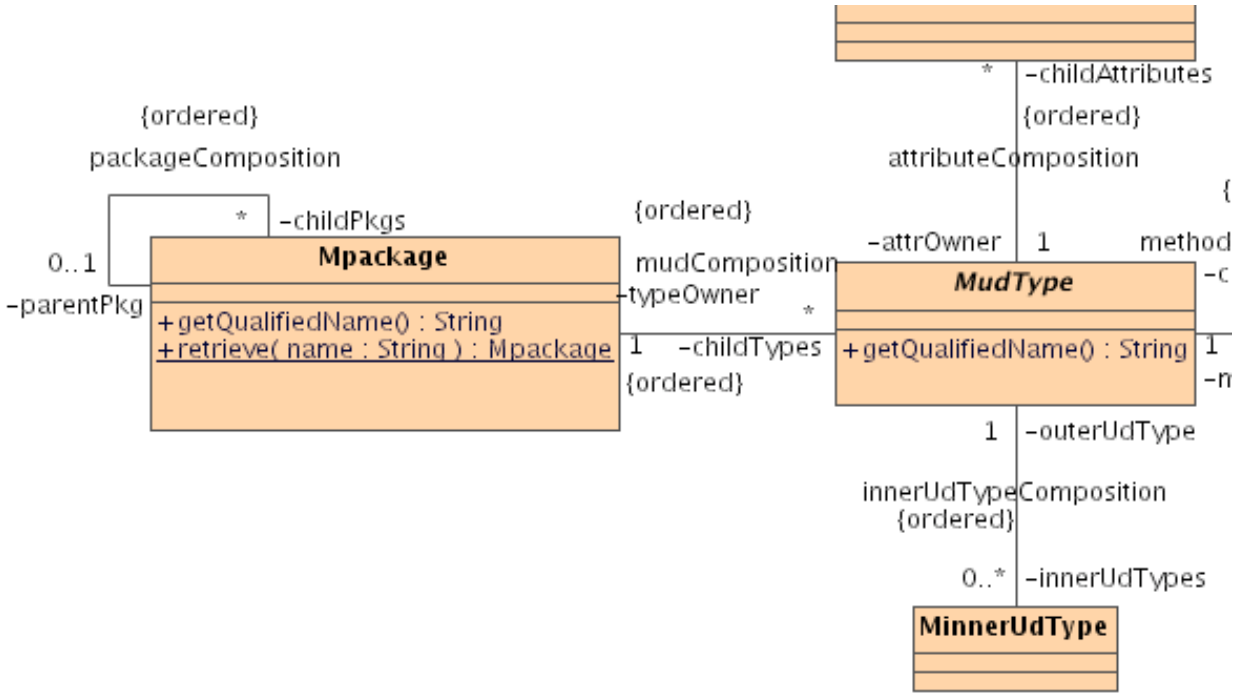
- Pieter Van Gorp, Hans Schippers, Olaf Muliawan
  - Query specification
  - Query generation

JDT2MDR

- Transformation between
  - AST's provided by Eclipse Java Development Tools
  - Grammy/FAMIX alike metamodel

- Goals:
  - Facilitate querying Java software systems
    - SDM, OCL, plain API usage
  - Cheap queries (yet expensive parsing)


Mattribute
MattributeReference
{ordered}
packageComposition
*
-childPkgs
{ordered}
attributeComposition
*
-childAttributes
{ordered}
attributeReferenceComposition
*
-childRefs
{ordered}
0..1
-parentPkg
Mpackage
+getQualifiedName() : String
+retrieve(name : String) : Mpackage
{ordered}
mudComposition
-typeOwner
*
1
-childTypes
{ordered}
-attrOwner
1
MudType
+getQualifiedName() : String
{ordered}
methodComposition
-childMethods
*
1
-methodOwner
Mmethod
+hasSimilarSignatureAs(otherMethod : Mmeth
1
-refOwner
1
-outerUdType
innerUdTypeComposition
{ordered}
0..*
-innerUdTypes
MinnerUdType
1
-parameterOw
parameterComposition
{ordered}
*
-childParamete
Mparameter


*
-childAttributes
{ordered}
{ordered}
packageComposition
attributeComposition
*
-childPkgs
{ordered}
0..1
-parentPkg
Mpackage
+getQualifiedName() : String
+retrieve(name : String) : Mpackage
mudComposition
-typeOwner
*
1
-childTypes
{ordered}
-attrOwner
1
MudType
+getQualifiedName() : String
1
-outerUdType
innerUdTypeComposition
{ordered}
0..*
-innerUdTypes
MinnerUdType

Collaboration benefits

LORE

- Application minded
  - refactoring candidates
  - link tests <-> code
- Reduced learning curve

FOTS

- Technique minded
  - graph transformation
  - code generation
- MotMot case study
- Beta-test data