

A Pragmatic Study of Binary Class Relationships

Recovering relevant UML-like class diagrams with Ptidej

Yann-Gaël Guéhéneuc

Hervé Albin-Amiot

guehene@iro.umontreal.ca, albin@emn.fr



École des Mines
de Nantes, France



Object Technology
International, Inc., Canada



Université de Montréal,
Canada



Objectives

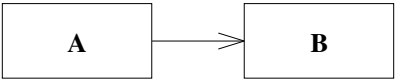
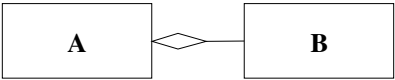

- Long term: Automated help to understand and to document object-oriented programs
 - Abstract implementation-level artefacts
- Short term: Produce relevant UML-like class diagrams and (design) pattern representations from source code



Needs

- To bridge the discontinuity between source code and class diagrams
 - Precise definitions of class-diagram artefacts: Class/interface, relationships
 - Models to describe source code and class-diagram artefacts
 - Static and dynamic analyses tools to generate the models

Relationships in a nutshell

-  **Association**
 - Says that instances of a class *A* send messages to instances of a class *B*
-  **Aggregation**
 - Association in which the whole (*A*) declares a part (*B*) (a field in the whole)
-  **Composition**
 - Aggregation in which the part (*B*) belongs exclusively to the whole (*A*)



Tools

- PDL¹: To describe source code, UML-like class diagrams, design patterns
- Caffeine: To obtain and abstract dynamic information from program executions
- Ptidej²: To serve as a front-end to other tools and to display class diagrams

1: Pattern (and Abstract-level) Description Language

2: Pattern Trace Identification, Detection, and Enhancement in Java

