A Transformation System for Java

Paulo Borba, Gustavo Santos, Adeline Sousa
Informatics Center
Federal University of Pernambuco
Recife, Brazil

Language specific transformation tools...

- implement only a fixed set of transformations

or

- allow the user to define new transformations through direct access to the transformation API
User-defined transformations

Using an extension of Java’s syntax with meta-programming constructs

- more accessible to Java programmers
- semantics-aware pattern-matching
- more expressive than templates

Transformation language

class #class [extends #superClass] {
    public #type #field;
    #fields;
    #methods;...
}

Meta-variable for class names
Meta-variable for field declarations
Optional matching
EJB pattern language

- Controller
- Remote Home
- Entity bean
- Entity class
- Controller
- Business collection
- Data interface
- Data collection
- Remote Home
- Entity bean
- Entity class

Product line of transformation systems

- TS Transformations
  - Language specific variations
  - Java TS
  - Haskell TS
  - Java TS
  - JSP TS
  - C# TS

Java TS
Conclusions

- Accessible and semantics-aware transformation languages
  - extending the syntax of the base language
- Low cost, and small development time, for language specific transformation systems
  - generation of transformation systems