A Pedagogic Programming Environment for Java that Scales to Production Programming

Charles S. Reis
Master’s Thesis Defense
Rice University
April 16, 2003
Pedagogic IDEs

• Useful tools in courses
  • Simple, easy to learn
  • Guiding philosophy for features

• Usually limited to intro level
  • Lack of powerful features
  • Restrictive interfaces (eg. UML)
Professional IDEs

- Many advanced features
- Large, Cumbersome
  - Significant overhead
  - Complex user interfaces
  - Not designed for students
- Avoided by many professionals!
Motivation

• Can pedagogic IDEs be useful at the level of production development?
DrJava

- Pedagogic IDE for intro level
  - Simple, intuitive
  - Interactive (REPL)
  - Focus on source language
DrJava Development

• Created by students at Rice
  • Object oriented, design patterns
  • Extreme Programming (XP)
  • Open Source

• Customers worldwide
Goal

- Extend DrJava to support production programming
  - Small set of new features
  - Ease transition to professional IDEs
  - Teach production programming with DrJava
Necessary Features

• REPL still useful
• Easy access to multiple files
• Traditional debugger
  • Suspend execution, query values
• Test-driven development
  • Integrated support for unit tests
Editing Multiple Files

- Java projects span many files
- Need convenient access to many classes at once
- Add document selector
Interactive Debugger

- Traditional features
  - Breakpoints
  - Stepping
  - Query values

- Integrated with Interactions Pane (REPL)
Debugger + REPL

• Flexible Points of Entry
  • Not just main method
  • Easily repeat experiments

• Interact with state in Java
  • Query, modify values
  • Call methods, etc
Unit Testing Support

• Key to incremental development
• Quality Safeguard

• Easy to write, run
  • JUnit framework
  • “Test” button
  • Visual feedback
Leveraging Professional IDEs

- Occasionally useful
  - Powerful refactoring tools
- Desire an easier transition from pedagogic IDEs
IBM’s Eclipse

• Widely used, open source IDE
• Everything is a “plug-in”
• Many advanced features
• Active ties with academia
DrJava Plug-in for Eclipse

• Innovation Grant from IBM
• Ease transition to Eclipse
  • Simplify user interface
  • Provide Interactions Pane (REPL)
  • Debugger + REPL

• REPL also useful for professional developers
Plug-in Development

• Code Re-use
  • All logic directly from DrJava
  • Single point of control for bug fixes, feature improvements

• Refactoring DrJava
  • More modular design
  • Safe and easy: unit tests!
DrJava’s Scalability

- DrJava team uses DrJava
  - Effective tool for its own development
- Scales to Production Programming
Teach Production Programming

- Use DrJava to teach production programming skills
  - Common, familiar environment
  - Select DrJava as course project!
Extend DrJava in a Course

- Students can:
  - Learn effective practices (XP)
  - Join an existing product team
  - Maintain a product
  - Support customers
Extreme Programming

• Expose students to effective development practices
  • Ubiquitous unit testing
  • Pair programming
  • On-site customer
  • Incremental releases
Classroom Challenges

• Time constraints
• Scarce resources

• How to:
  • Quickly transfer knowledge?
  • Adapt Extreme Programming?
  • Manage development?
XP: Knowledge Transfer

• Pair Programming
  • With experienced TAs
  • With other students

• Unit Tests
  • Executable documentation
Adapting XP for Classroom

• Pair Programming
  • Lab time, students select own pairs

• On-site Customers
  • Students themselves (using DrJava)

• No fixed deadlines
  • Require 10 logged hours per week
Managing Development

• TA’s as Project Managers

• SourceForge.net
  • Free open source project hosting
  • Professional quality management
    • Bug reports
    • Feature requests
    • Task management
Case Study: COMP 312

• Early unit test assignment
• Bug fixes
• Large features in small tasks
  • 2002: JUnit support, debugger, configurability
  • 2003: Interactive debugger, Javadoc, interactions pre-processor
Results

• DrJava effective for production programming
  • Used in its own development
• Eclipse plug-in eases transition
• Excellent results from 312
  • Many core features implemented
  • Students exposed to process
Conclusion

• Pedagogic IDEs can scale to production development
  • DrJava’s simplicity preserved
  • Effective for large projects
  • Useful for teaching production programming skills