Refactoring using Together

Amos You, Principal Consultant
Borland Software China

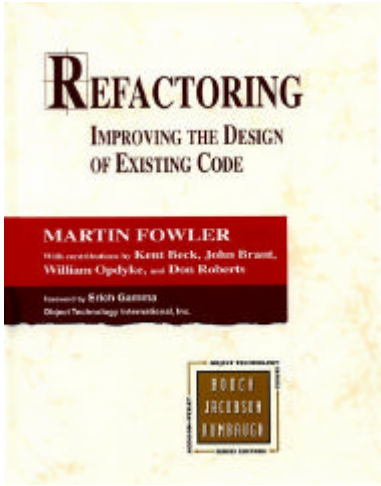
议程

- Refactoring的相关概念
  - 定义，目标，核心内容，基本技巧，典型困难
- Together为Refactoring提供综合支撑环境
  - LiveSource Technology, Metrics, Refactoring Facilities, JUnit Framework, Version control
- 用Together有效进行典型的Refactoring工作
- 建议
  - 理解方法，掌握工具，加速成功

Refactoring的相关概念

Refactoring的相关概念

- 定义
  - Martin Fowler defines refactoring as: “…the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure.”
  - A frequent result of code reviews
- 目标
  - To improve design, understandability and code quality
  - To decrease the cost of maintenance



Refactoring的相关概念

- 核心内容
  - 22 “bad smell” (into 7 categories), The Reasons
  - 72 “Refactoring”, The corresponding Actions
- 基本技巧
  - Taking small steps
  - Testing often

Refactoring的相关概念

- 典型困难
  - 代码的直观性和概括性
  - 发现 “bad smell” 的手段
  - 局部改动的整体完整性
  - 有效的集成单元测试设施
  - 对代码调整的阶段可逆性保障

Together为Refactoring提供综合支撑环境

Live Source Technology

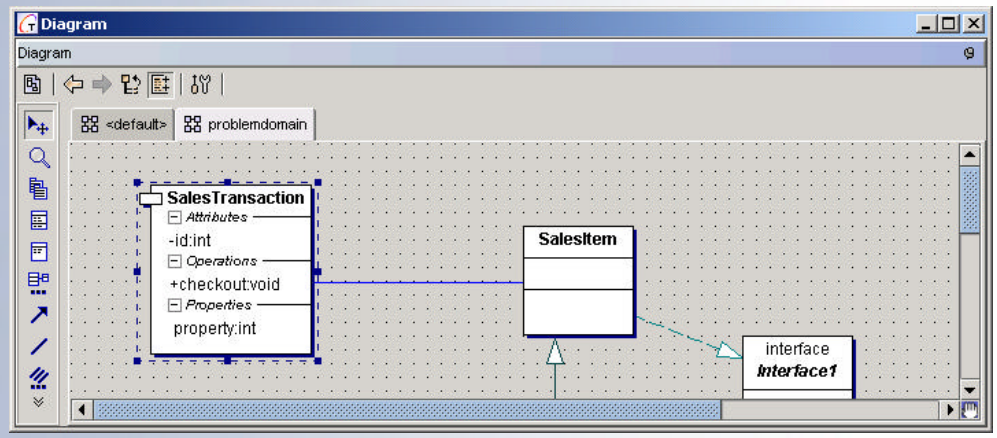
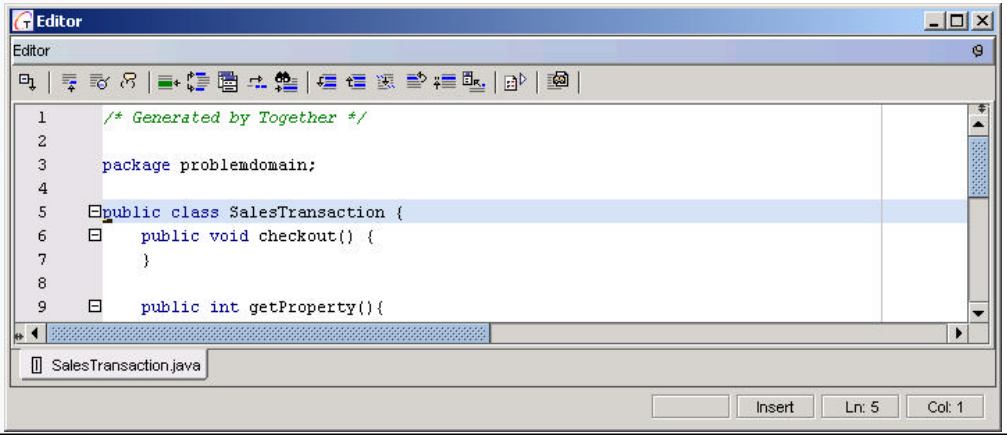
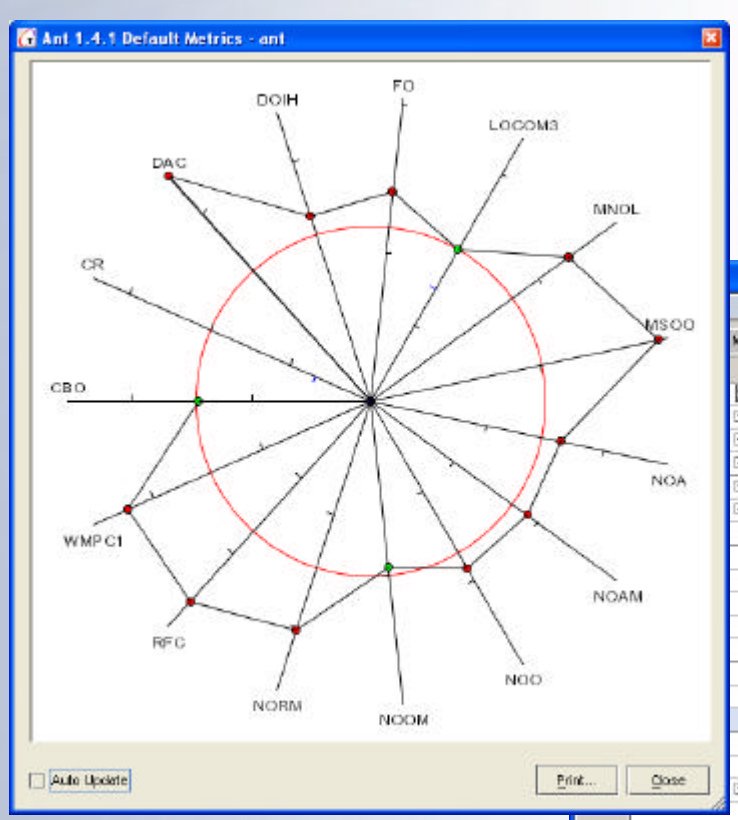


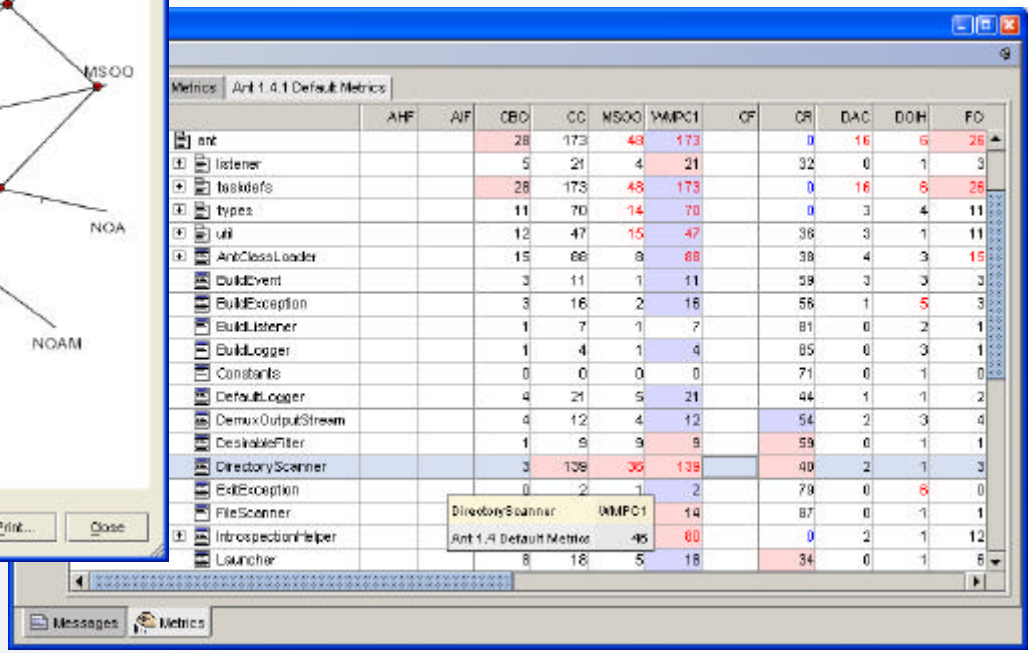
Diagram elements generates source

Source reflected in diagram

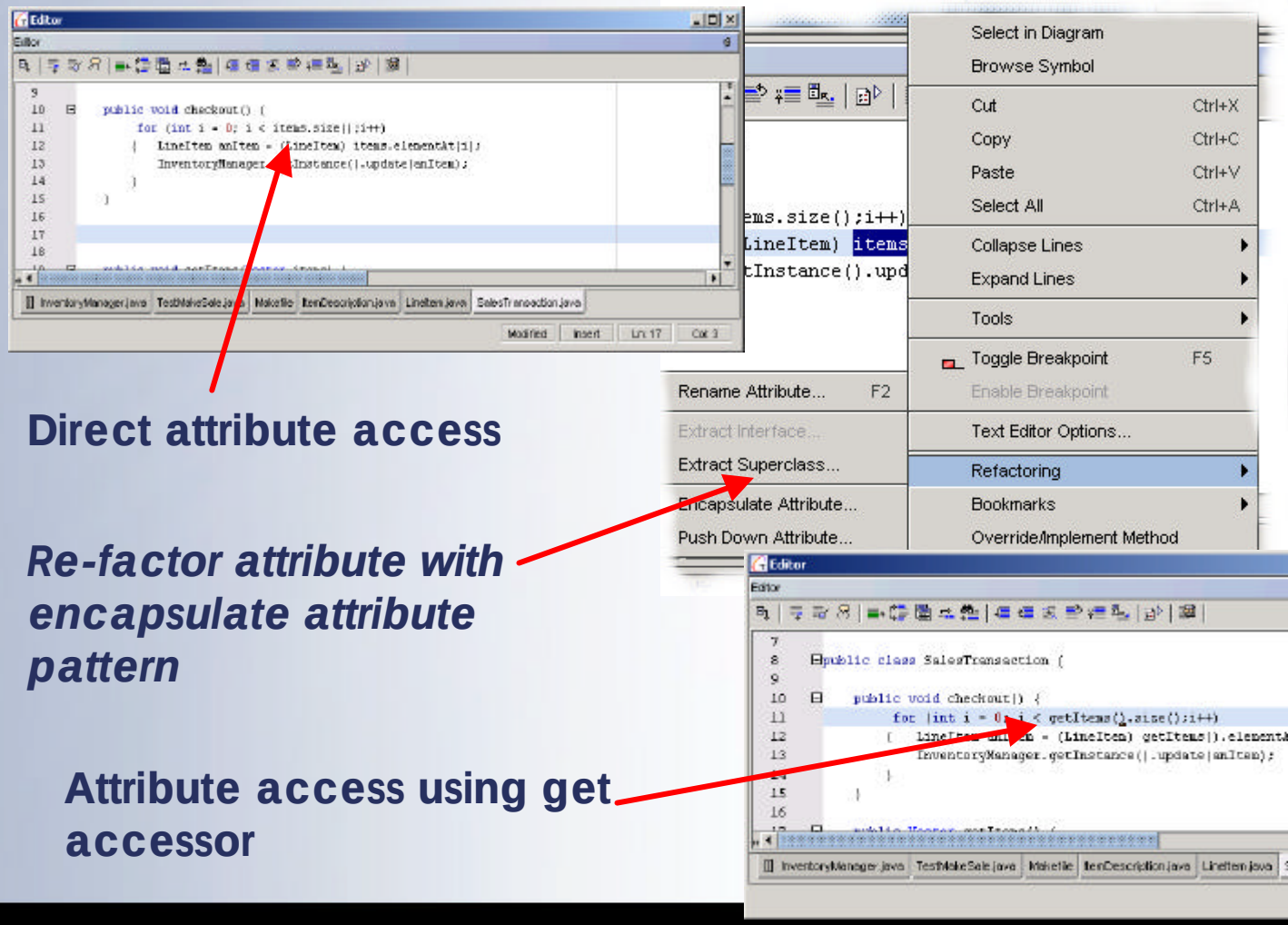


Metrics

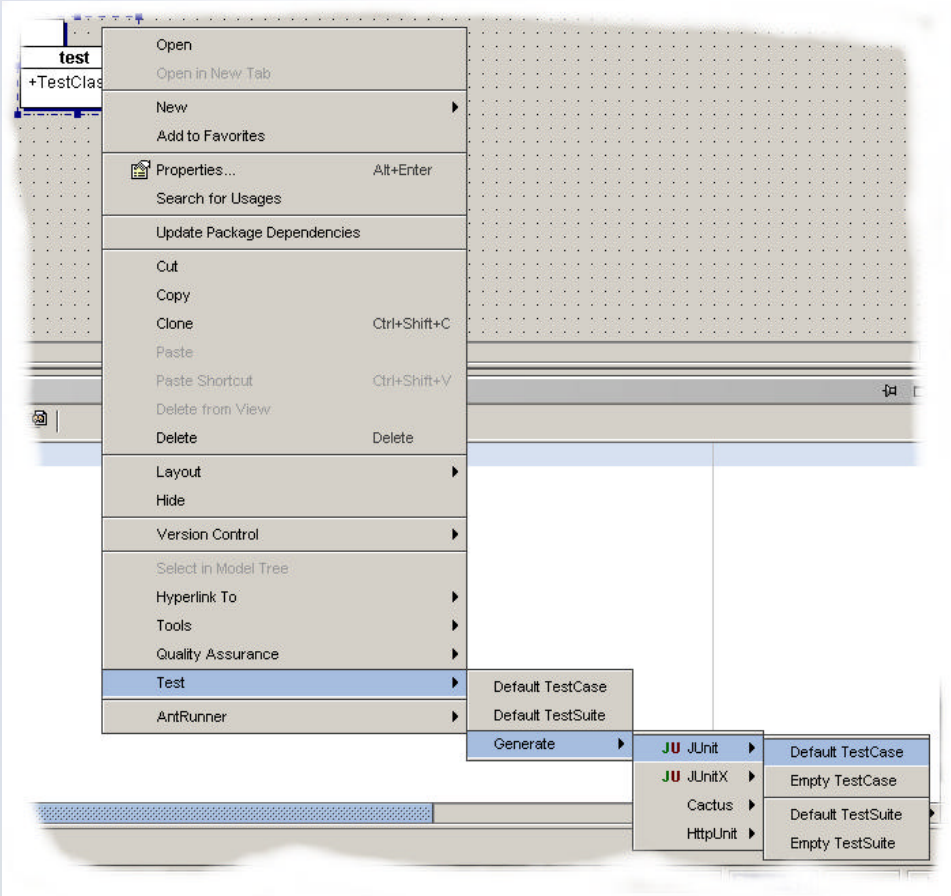

Ant 1.4.1 Default Metrics - ant
DOIH
FO
LOCOM3
DAC
MNOL
CR
MSOO
CBO
NOA
WMPC1
NOAM
RFC
NOO
NORM
NOOM
Auto Update
Print...
Close


Metrics
Ant 1.4.1 Default Metrics
AHF
AIF
CBO
CC
MSOO
WMPC1
CF
CR
DAC
DOIH
FO
ant
listener
taskdefs
types
util
AntClassLoader
BuildEvent
BuildException
BuildListener
BuildLogger
Constants
DefaultLogger
DemuxOutputStream
DesirableFilter
DirectoryScanner
ExitException
FileScanner
IntrospectionHelper
Launcher
DirectoryScanner
WMPC1
Ant 1.4 Default Metrics
Messages
Metrics

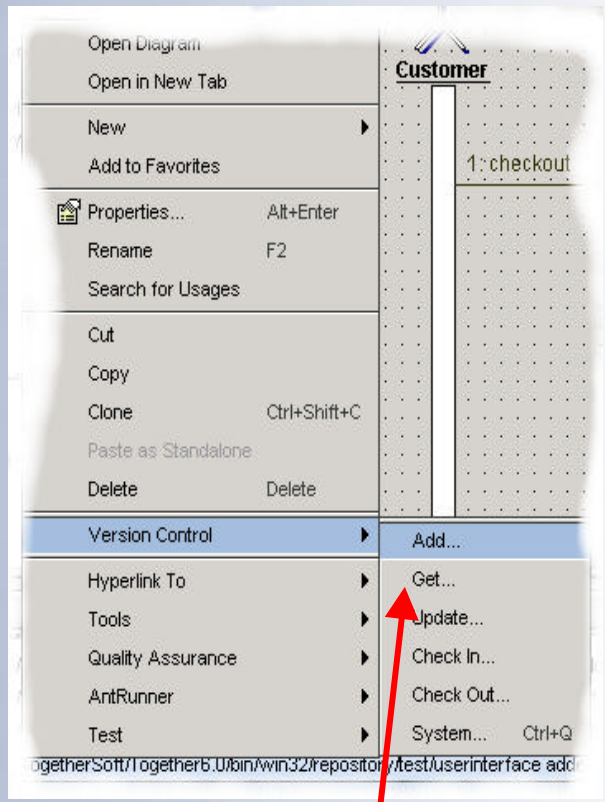
Refactoring Facilities

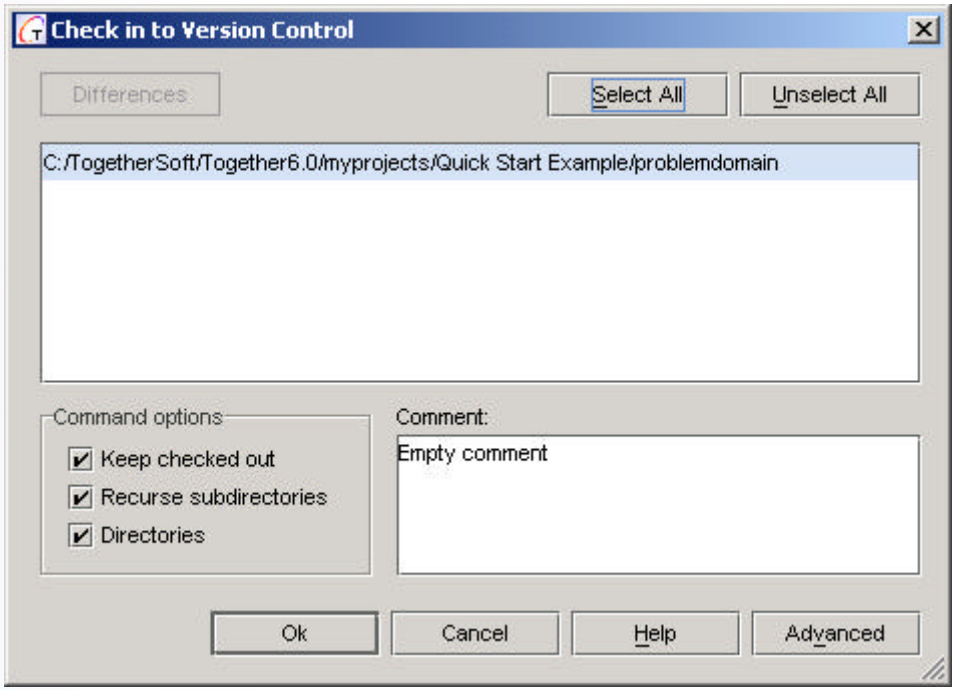


JUnit Framework



Version Control





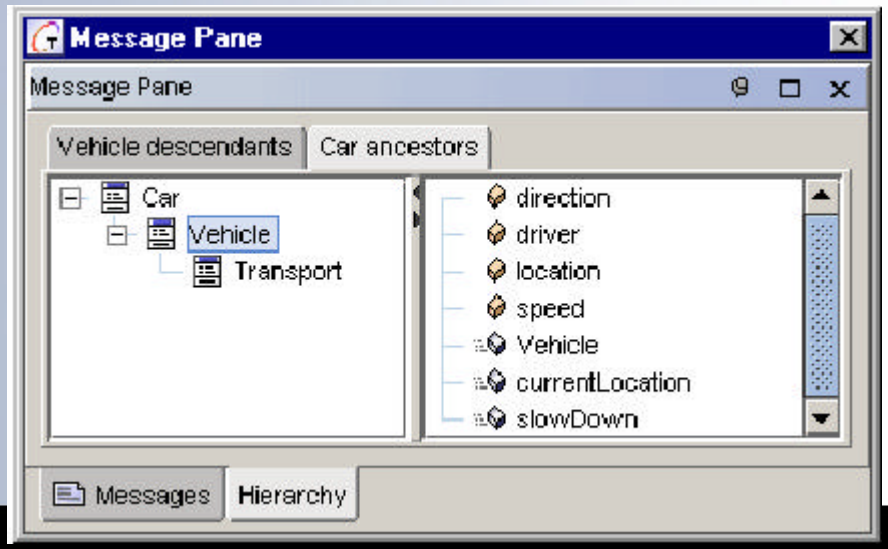
Version control operations

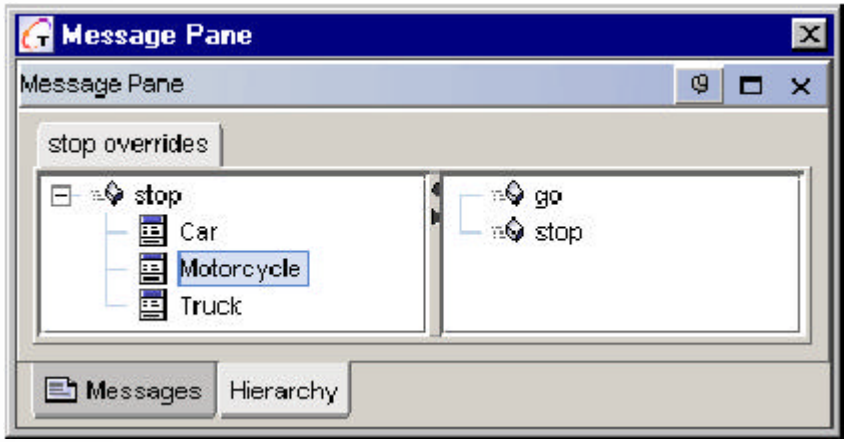
Check in dialog

用Together有效进行典型的Refactoring工作

Showing code structure

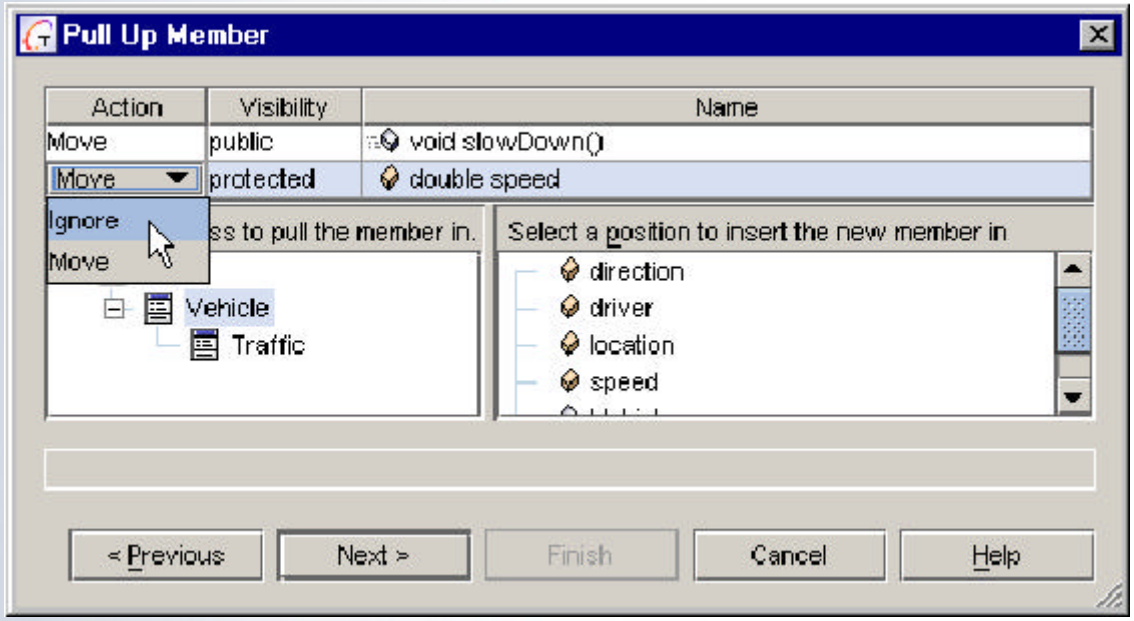
- Show Ancestors
- Show Descendants
- Show Implementing Classes
- Show Overrides



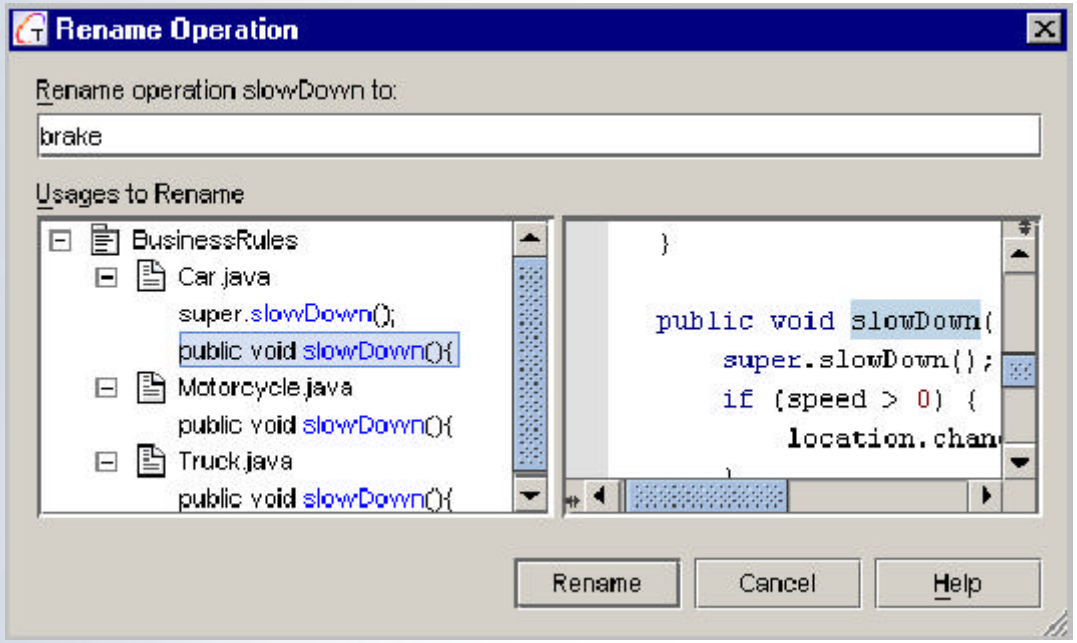


Moving classes, interfaces, attributes, and operations

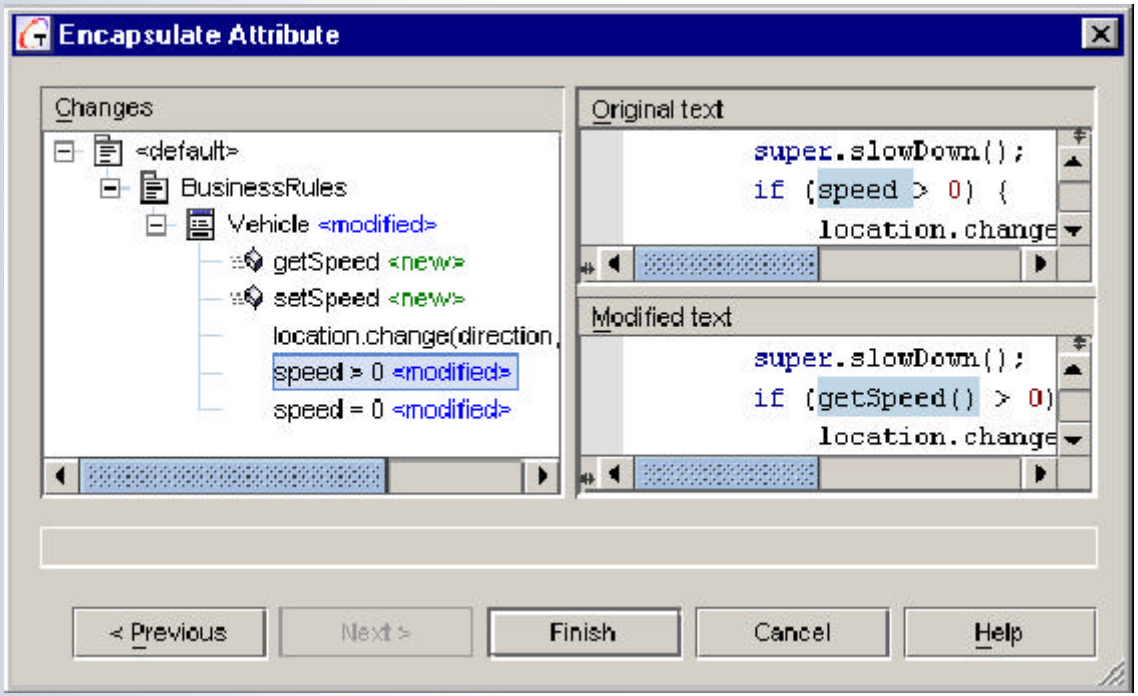
- Moving classes and interfaces
- Moving attributes and operations in the class hierarchy



Renaming

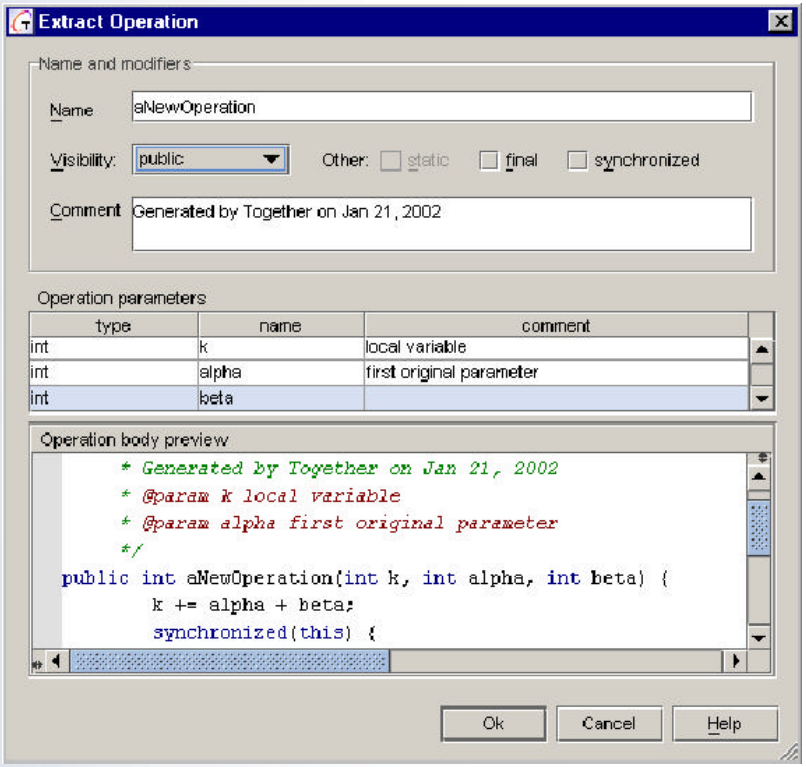

Rename Operation
Rename operation slowDown to:
brake
Usages to Rename
BusinessRules
Car.java
super.slowDown();
public void slowDown(){
Motorcycle.java
public void slowDown(){
Truck.java
public void slowDown(){
}
public void slowDown(
super.slowDown();
if (speed > 0) {
location.chan
Rename
Cancel
Help

Encapsulating attributes



Extracting interfaces, superclasses, and operations

- Extracting interfaces and superclasses
- Extracting operations



建议

建议

- 理解方法
  - Basic catalogue available as book ”Refactoring: Improving the design of existing code”, Addison-Wesley, 1999
  - Extended catalogue available online http://www.refactoring.com/
- 掌握工具
  - 下载工具: www.borland.com
  - 快速入门: Together Practical Guide
  - White paper: Software Remodeling: Improving Design and Implementation Quality
- 加速成功
  - 选择Borland专业咨询与培训服务

谢谢

amos.you@borland.com