
Towards a Refactoring Benchmark

Serge Demeyer

Lab on Reengineering (LORE)

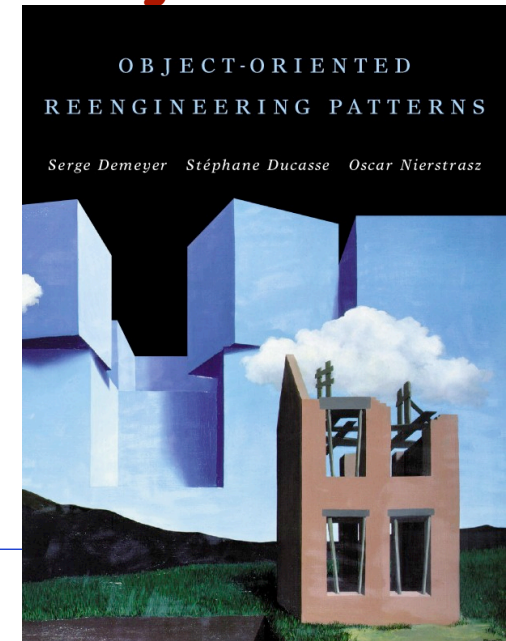
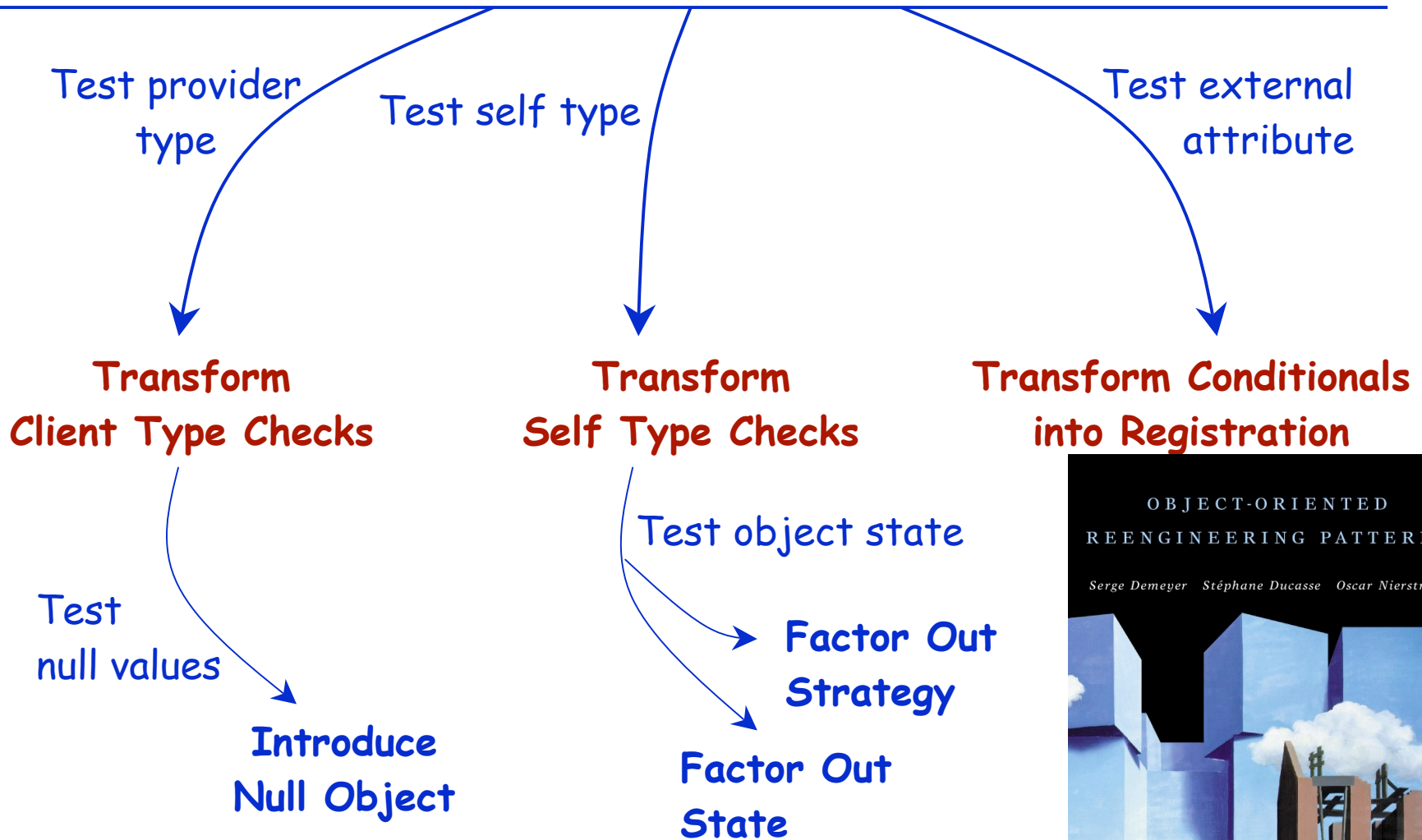
University of Antwerp

Presentation for the ELISA - Workshop

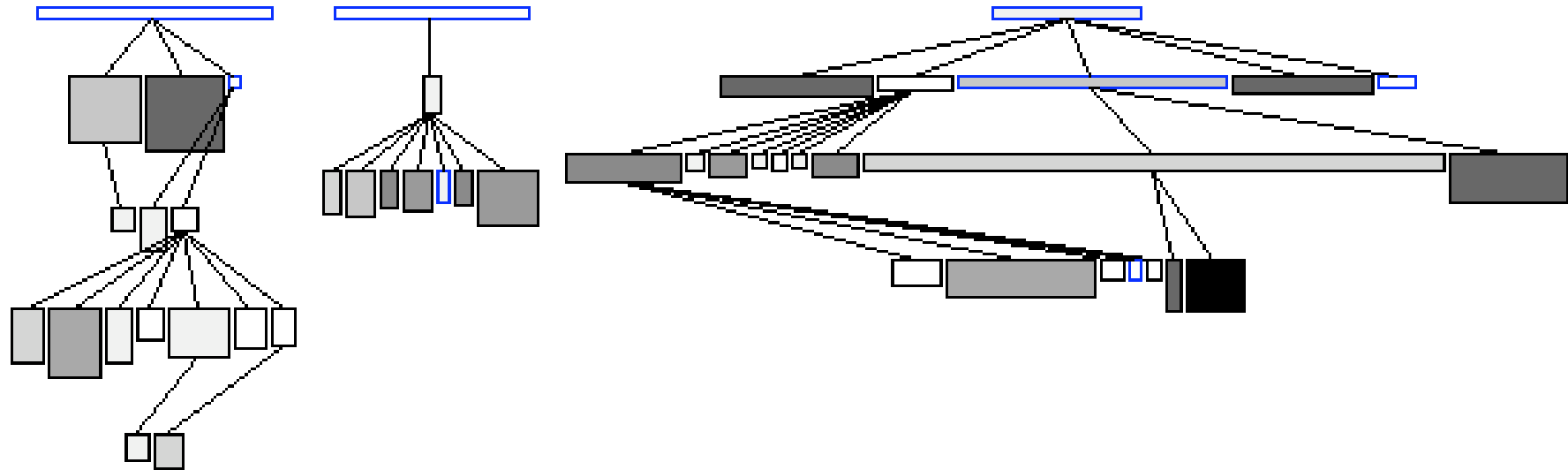
(September 2003, Amsterdam - The Netherlands)



Story #1

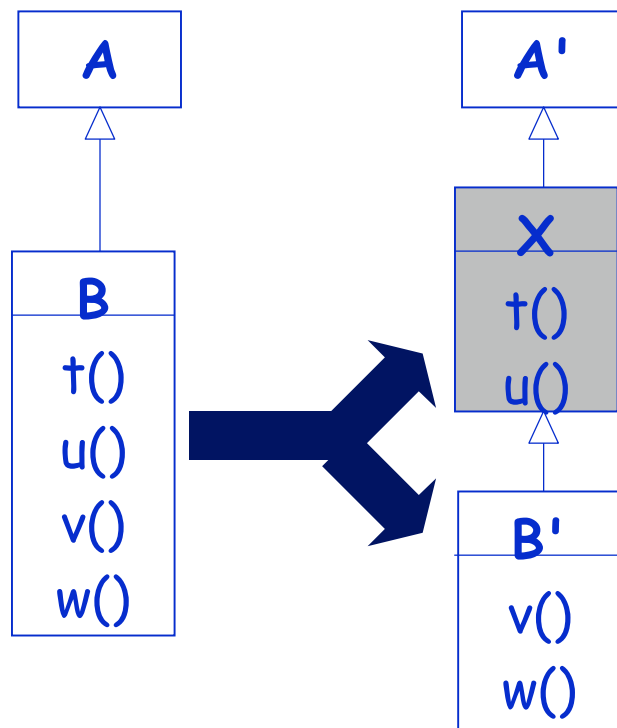


Story #2



Boxes:	Classes
Width:	# methods added
Height:	# methods overridden
Color:	# method extended

Story #3



Split B into X and B'

/ Hierarchy nesting level increased */
($\text{delta_HNL}(B') > 0$) and*

/ Number of methods decreased */
($(\text{delta_NOM}(B') < 0)$)*

/ Number of attributes decreased */
or ($\text{delta_NOA}(B') < 0$)*

Classification

Curative

(i.e. Which refactorings are good ?
How do tools support refactoring ?)



Predictive

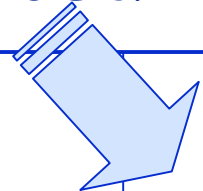
(i.e. Where to apply
Which Refactoring ?)



Retrospective

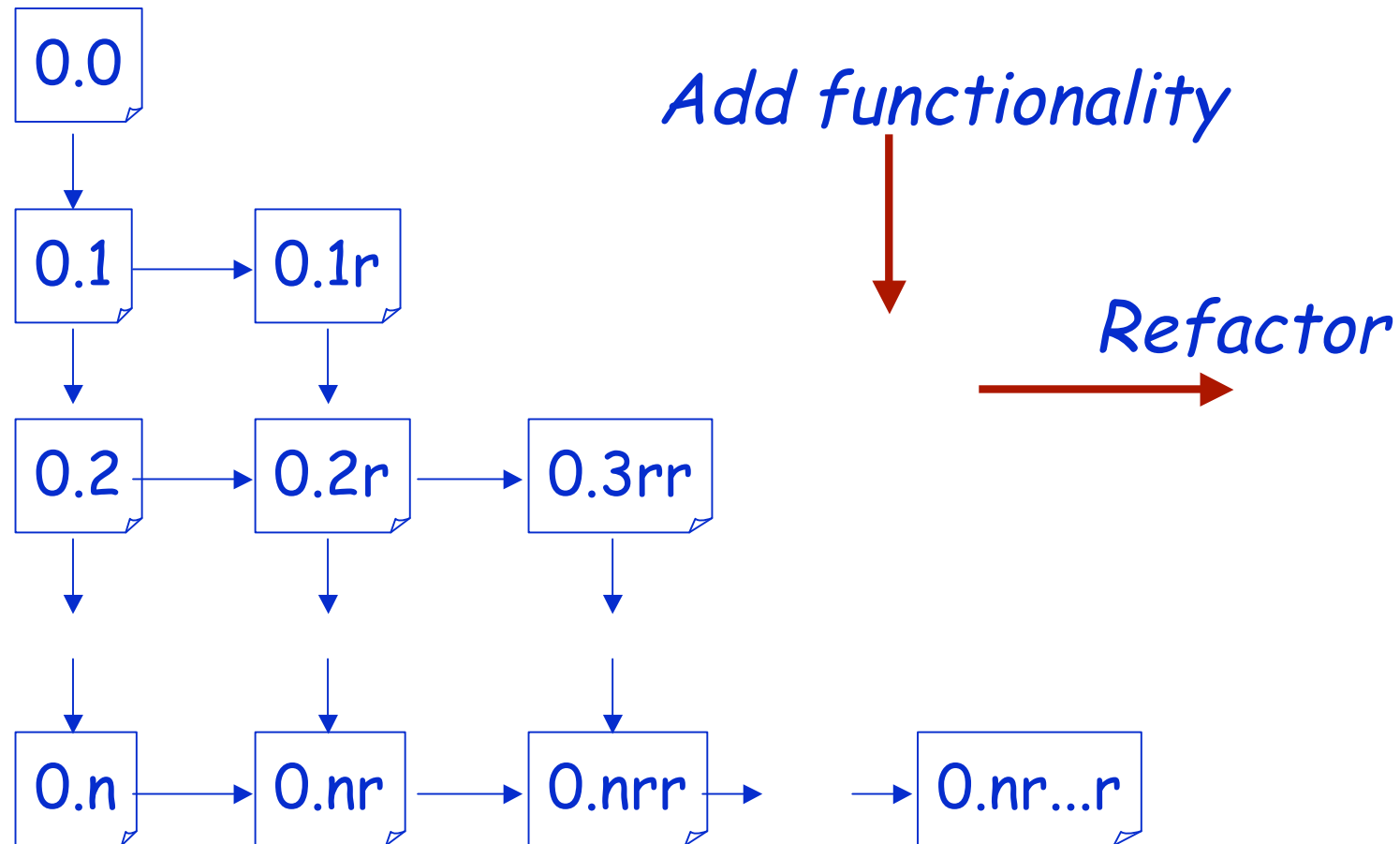
(i.e. Which Refactorings
have been Applied ?)

Benchmark proposal

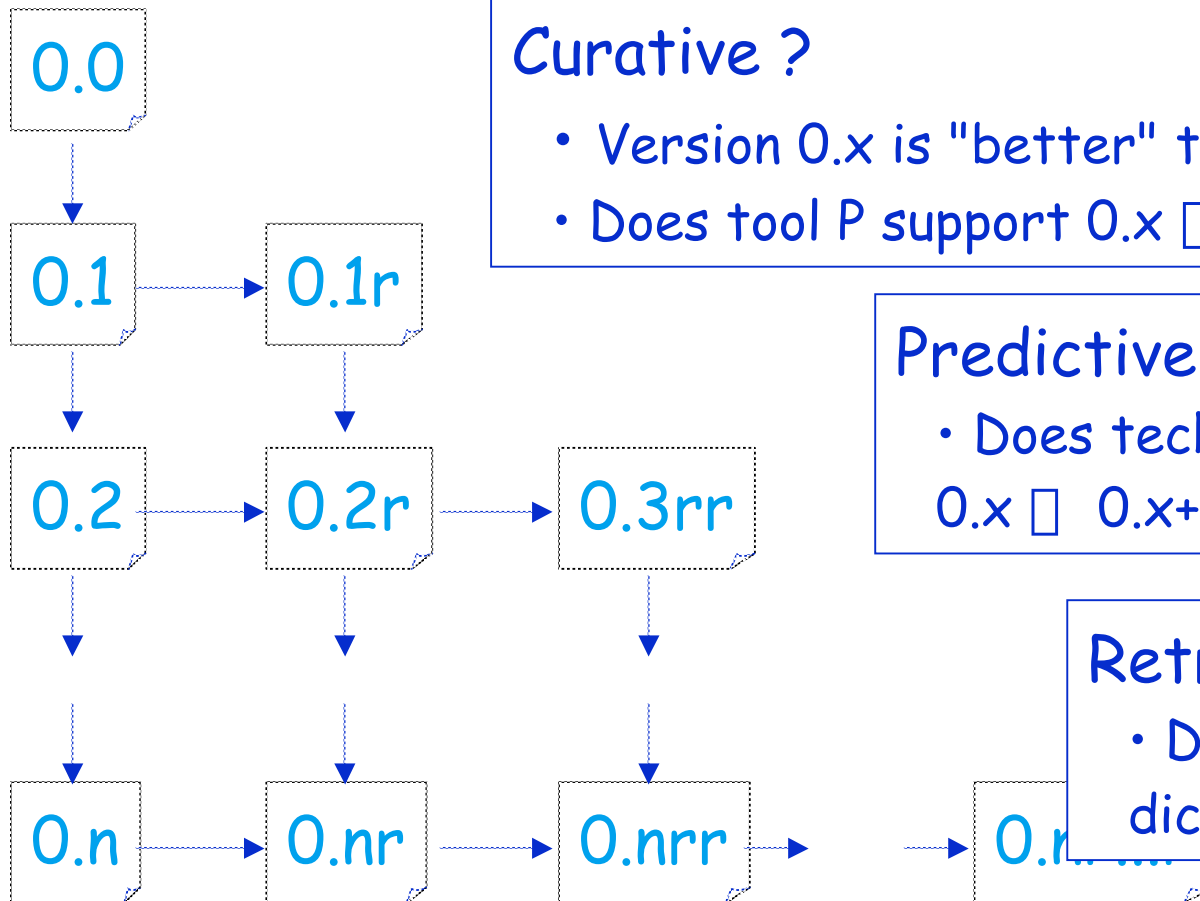


Case studies <ul style="list-style-type: none"> • Toy Example (LAN -Simulation) • Industrial System (VisualWorks & Swing) • Public Domain (HotDraw & ET++) • Open-source (Mozilla) 				
Characteristics <ul style="list-style-type: none"> • Life Cycle (analysis, design, ...) 				
<ul style="list-style-type: none"> • Evolution (scale, #iterations, ...) 				
<ul style="list-style-type: none"> • Domain (problem, solution, ...) 				

Case Study: LAN Simulation



Case Study: LAN Simulation



Curative ?

- Version 0.x is "better" than version 0.x-1 ?
- Does tool P support 0.x \square 0.x+1 ?

Predictive ?

- Does technique Q predict 0.x \square 0.x+1 ?

Retrospective ?

- Does technique R discover 0.x \square 0.x+1 ?

Discussion

- Does it makes sense to work out this LAN benchmark ?
 - Would you use it ? yes no