Legacy Software Restructuring: Analyzing a Concrete Case

Nicolas Anquetil, Jannik Laval
RMoD team
INRIA / Univ. Lille-1

Agenda

- Software Restructuring
- Cohesion/Coupling dogma
- Experiment idea
- A case study: Eclipse RCP
- Experiment set-up
- Results
- Conclusion

Software Restructuring

- Software systems evolve, their structure (architecture) deteriorates
- How can we help?
  - Metrics to evaluate the quality of the architecture
  - Tools to restructure (optimization of the quality metrics)

Software Restructuring

- Software systems evolve, their structure (architecture) deteriorates
- How can we help?
  - **Metrics** to evaluate the **quality** of the architecture
  - Tools to restructure (optimization of the **quality metrics**)

Cohesion/Coupling dogma

- Quality of modularization boils down to

  High cohesion & Low coupling

  (a module should be highly cohesive, and poorly coupled)

- Initially: semantic cohesion/coupling
- But for facility reasons, we measure syntactic cohesion/coupling

Cohesion/Coupling dogma

- Are we so sure that

  High cohesion & Low coupling

  is a good idea?
  - [Abreu, Goulão, CSMR'01]
  - [Bhatia, Singh, SERP'06]
  - [Sindhgatta, Pooloth, COMPSAC'07]

- What proof do we have?

Experiment idea

- Test the validity of

  High cohesion & Low coupling

  on a modularization of know value

Experiment idea

- Test the validity of

  High cohesion & Low coupling

  on a modularization of know value

- Problem: Only one theoretical known value for cohesion/coupling: 0
- Solution: Compare two values with known difference

Experiment idea

- We need real cases of explicit, successful, pure re-structuring efforts
  - Measure cohesion/coupling before
  - Measure cohesion/coupling after
  - Compare: Did it improve?
- Hypothesis: After an explicit, successful, pure re-structuring effort, cohesion/coupling of the system should improve

Experiment idea

- We need **real cases** of explicit, successful, pure re-structuring efforts
  - Need access to source code to evaluate (syntactical) cohesion/coupling
  - Need access to code before and after re-structuring effort
  - Seems easy: Open-source systems typically use some Version Control Systems

Experiment idea

- We need real cases of **explicit**, successful, pure re-structuring efforts
  - Used Google CodeSearch, not so easy
  - Very little efforts are documented as “re-structuring” in the wild
  - *(May be you can help?)*

Experiment idea

- We need real cases of explicit, **successful**, pure re-structuring efforts
  - Hypothesis: Proof of time

Experiment idea

- We need real cases of explicit, successful, **pure** re-structuring
  - No other activity on the system at the same time
  - Impossible to find in real life: Systems need to evolve
  - Threat to validity

A case study: Eclipse RCP

- Eclipse v2.1 → v3.0 (in 2004)
  - v2.1: Extensible IDE
  - V3.0: Rich Client Platform
- Also v2.0.1 → v2.1
  - Preliminary restructuring
    "Prior to 2.1, the org.eclipse.ui plug-in was the monolithic implementation of the Eclipse Platform UI. The above picture reflects the restructuring that done for 2.1 [...]"
- Also v3.0 → v3.1
  - Check, just after big restructuring

Experiment set-up

- Four successive versions of “core” Eclipse
- Metrics
  - Descriptive:
    #packages, #plugins, #classes, #methods, #method invocations, LOC
  - Cohesion/coupling:
    Bunch, Efferent/Afferent coupling (Ce/Ca)
  - *Cyclic dependencies (not shown here)*

Results

- Descriptive statistics

	#pckgs	#plugins	#class	#meth	#invoc	LOC
v2.0.1	101	10	3.209	23.172	53.302	417.109
v2.1	144	18	4.034	29.098	66.806	540.948
v3.0	251	26	6.449	44.377	100.667	804.071
v3.1	307	26	7.612	52.369	115.541	969.078

Results

- Bunch cohesion/coupling on packages

	Cohesion			Coupling		
	incr.	same	decr.	incr.	same	decr.
2.0.1 → 2.1	16	34	44	23	12	59
2.1 → 3.0	32	49	58	48	21	70
3.0 → 3.1	64	78	98	115	28	97

Results

- Bunch cohesion/coupling on packages

	Cohesion			Coupling		
	incr.	same	decr.	incr.	same	decr.
2.0.1 → 2.1	**16**	34	**44**	23	12	59
2.1 → 3.0	32	49	58	48	21	70
3.0 → 3.1	64	78	98	115	28	97

Results

- Bunch cohesion/coupling on packages

	Cohesion			Coupling		
	incr.	same	decr.	incr.	same	decr.
2.0.1 → 2.1	**16**	34	**44**	**23**	12	**59**
2.1 → 3.0	**32**	49	**58**	**48**	21	**70**
3.0 → 3.1	**64**	78	**98**	**115**	28	**97**

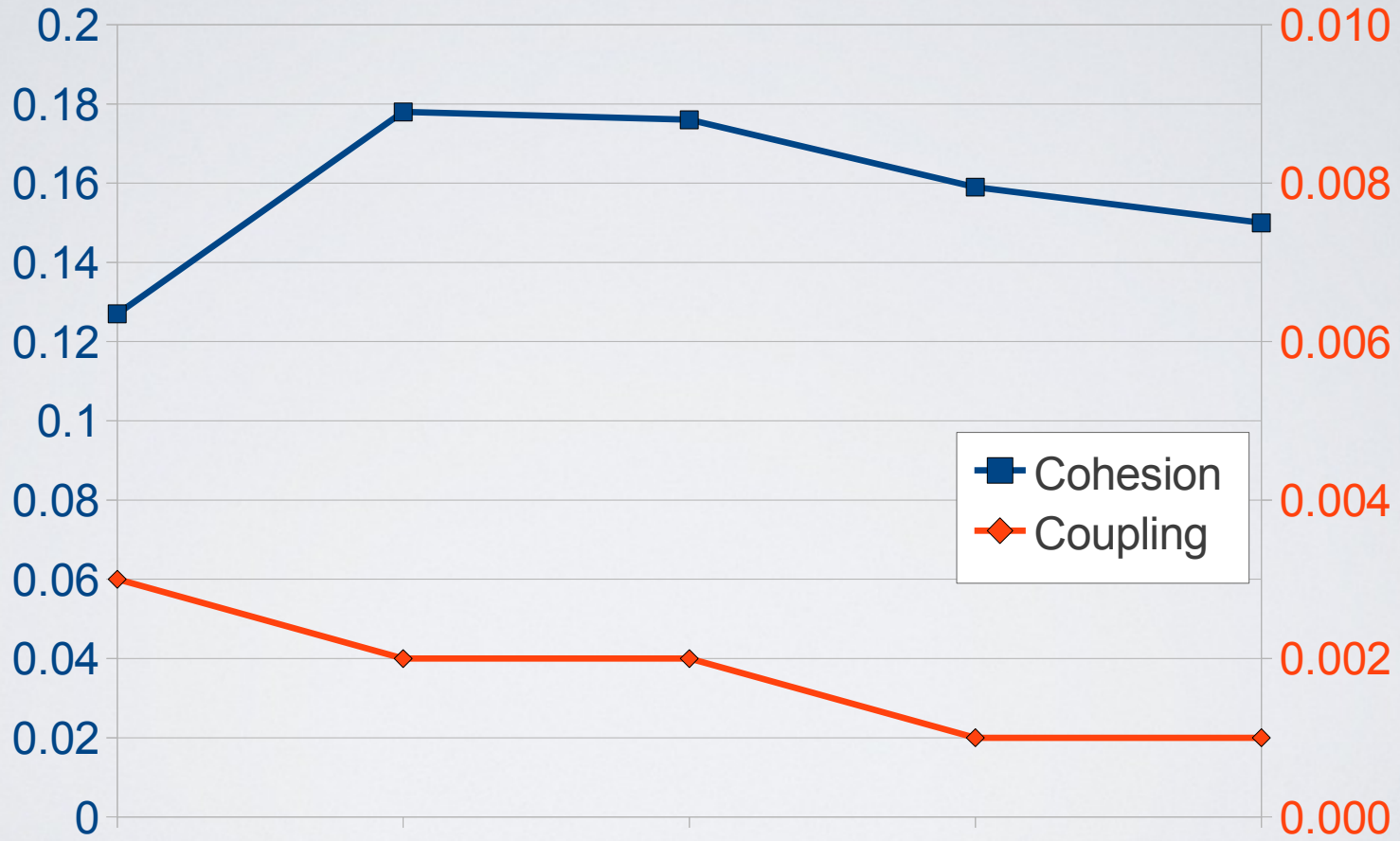
Results

- Efferent/Afferent coupling on packages

	Ce			Ca		
	incr.	same	decr.	incr.	same	decr.
2.0.1 → 2.1	52	33	13	58	26	14
2.1 → 3.0	75	43	25	88	38	17
3.0 → 3.1	119	72	53	124	79	41

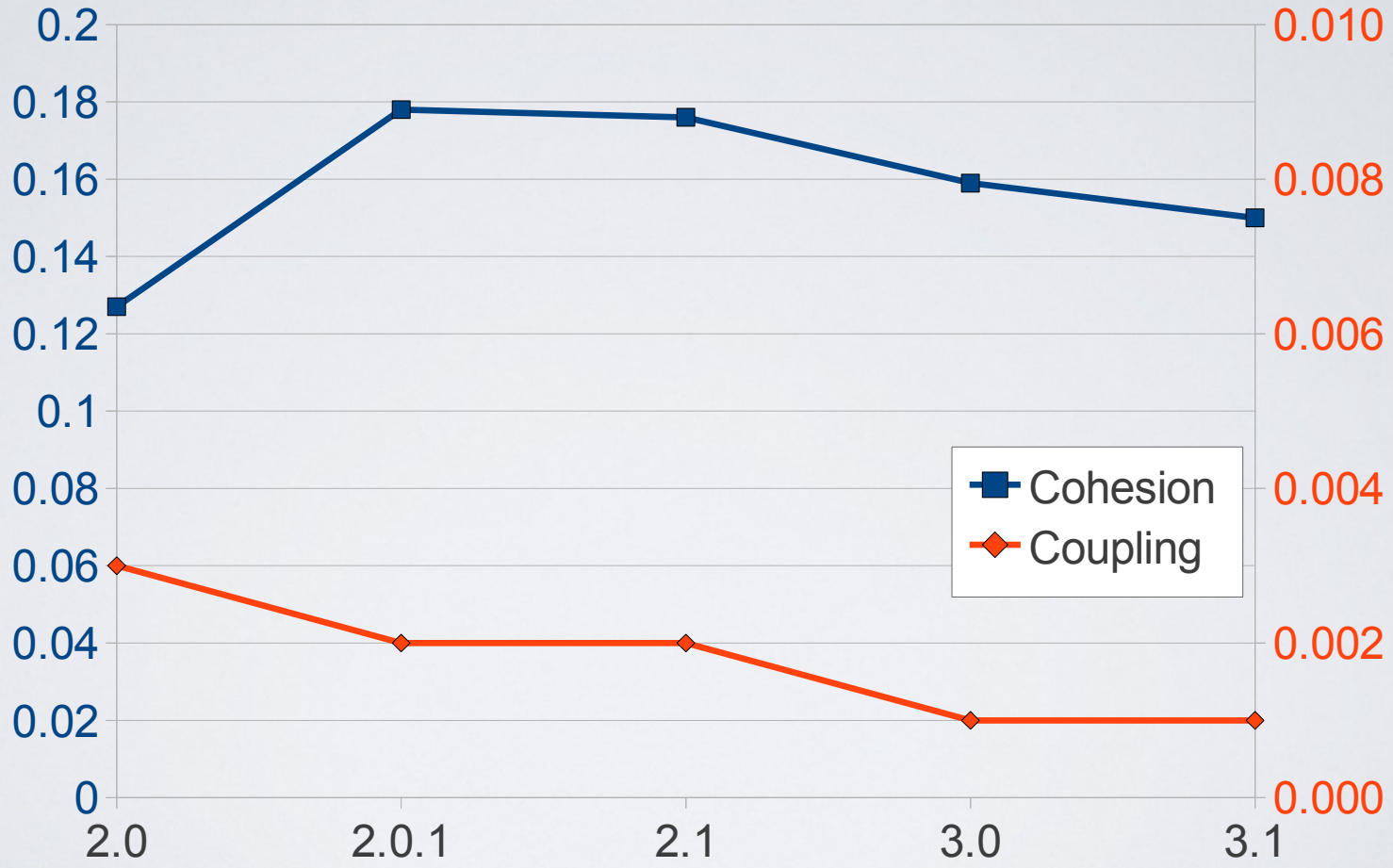
New Data (not in the paper)

- Eclipse, 5 versions



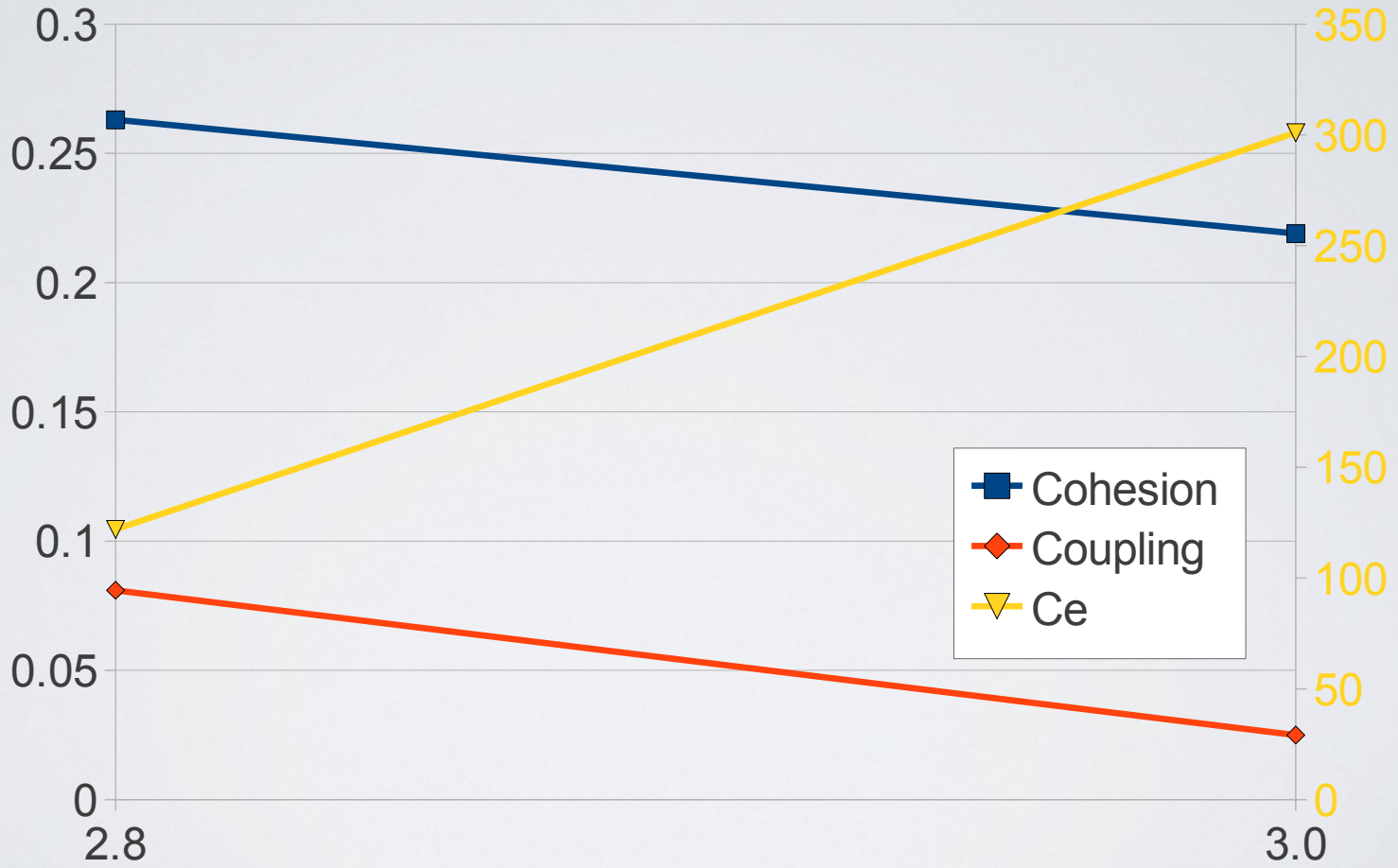
New Data (not in the paper)

- Eclipse, 5 versions



New Data (not in the paper)

- Seaside 2.8 → 3.0



Conclusion

- Cohesion/Coupling did not improve during 2 re-structuring efforts on Eclipse
  - Also Cohesion/Coupling seem to evolve jointly not oppositely
- Existing (tested) cohesion/coupling metrics do not measure what we want

- Need more experiments with more case studies