EXPERIENCES IN REPLICATING AN EXPERIMENT ON COMPARING STATIC AND DYNAMIC COUPLING METRICS

Richard Müller, Dirk Mahler, and Christopher Klinkmüller

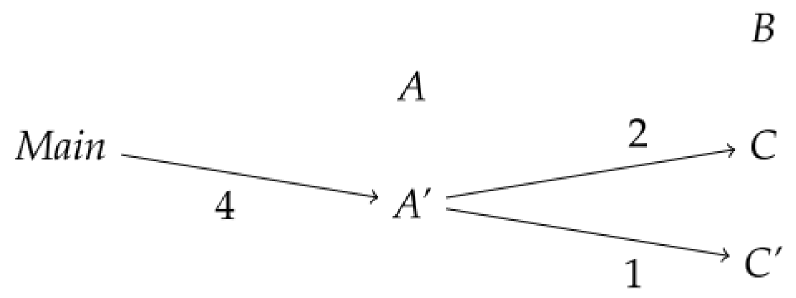
12th Symposium on Software Performance
Leipzig, November 9th, 2021

Outline

- Terms and Definitions
- Original Experiment
- Replication
- Challenges
- Conclusion

Coupling Metrics

- Coupling metrics (degrees) can be derived from a dependency graph
  - Nodes are program modules that are either **classes** or **packages**
  - Edges are call relationships between the modules and may have weights representing the number of calls (**weighted** or **unweighted**)
- The direction of calls is distinguished into outgoing and incoming calls referred to as **import** and **export** coupling degree of a program module and the sum of import and export coupling is referred to as **combined** coupling degree
- Depending on whether the connections were derived via static analysis of source and/or byte code, or via dynamic analysis of monitoring logs, coupling degrees are referred to as **static** or **dynamic**



class	import	export	combined
Main	4	0	4
A‘	3	4	7
C	0	2	2
C‘	0	1	1

Kendall-Tau Distance

- Kendall-Tau distance (τ) is a method to compare the difference between two linear orders (<1, <2)
- τ is between 0 and 1
  - If τ = 0: <1 is identical to <2
  - If τ = 1: <1 is the reverse of <2

[Briand, Emam & Morasca 1996]

Original Experiment

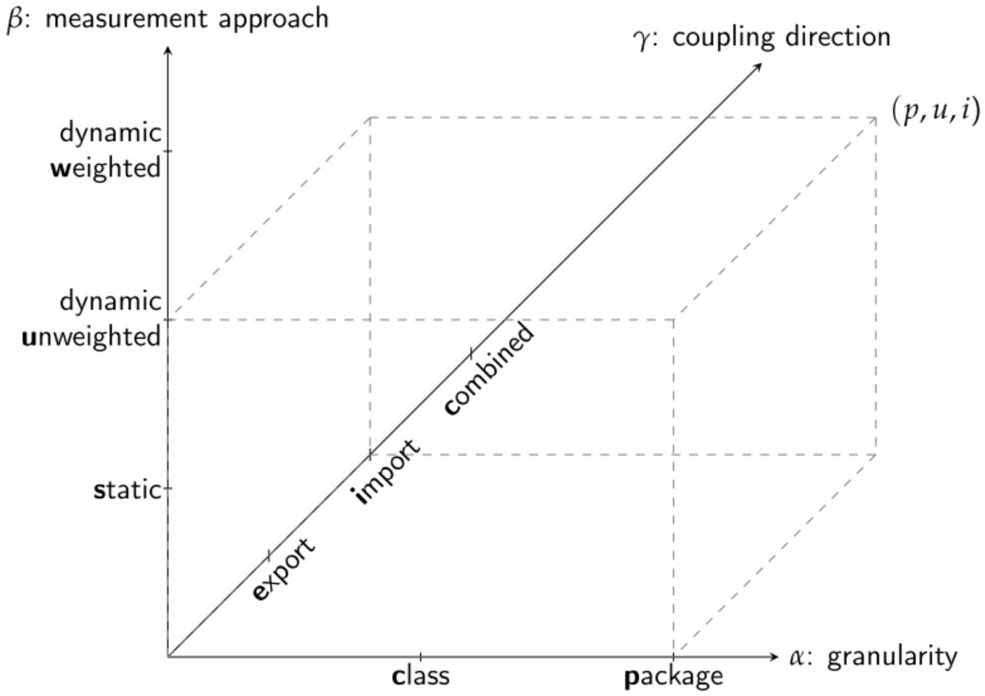
- Schnoor and Hasselbring empirically investigated the relationship between static and dynamic coupling metrics based on a series of four experiments

#	Date	Users	Method Calls	Jira version
1	February 2017	19	196,442,044	7.3.0
2	September 2017	48	854,657,027	7.4.3
3	February 2018	16	475,357,185	7.7.1
4	September 2018	58	2,409,688,701	7.7.1

[Schnoor & Hasselbring 2020]

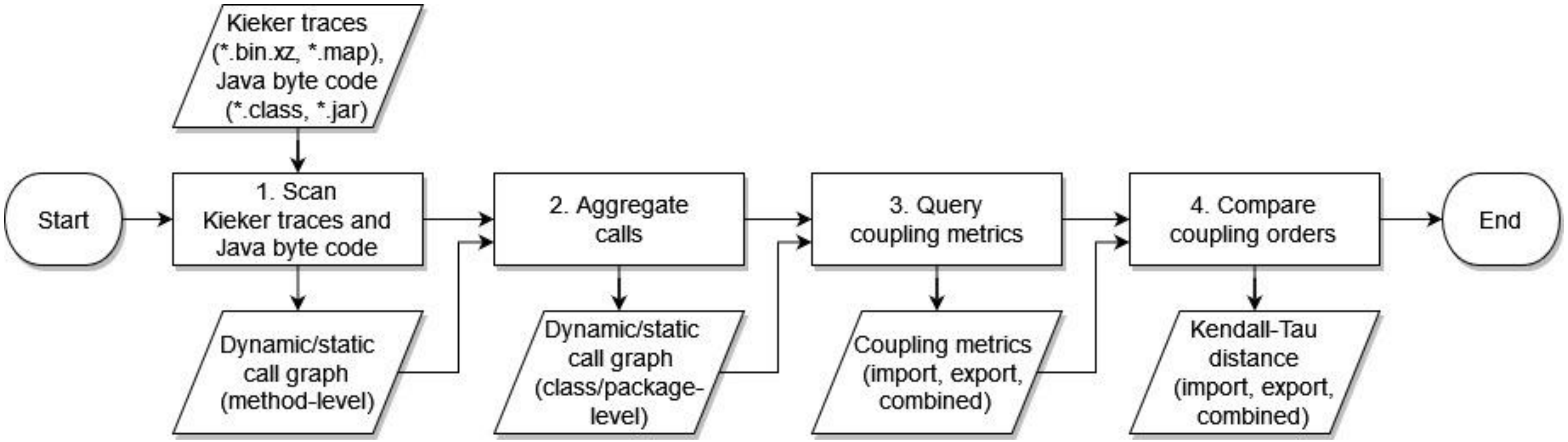
Original Experiment

1. The static and dynamic dependency graphs based on method calls were created for each experiment
   – BCEL: Byte code scan of method calls
   – Kieker: Monitored method calls at runtime
2. 18 different coupling metrics were derived from these graphs
   – Granularity: package and class level
   – Coupling direction: import, export, combined
   – Measurement approach: static, dynamic unweighted, dynamic weighted
3. The differences of the coupling metrics were studied by comparing the ranking obtained by ordering the program modules by their coupling degree using the Kendall-Tau distance



[Schnoor & Hasselbring 2020]

Replication



[Müller et al. 2018, Müller & Fischer 2019, Müller & Strempel 2020]

Results

- The results are presented using the following triple $\alpha : \beta 1 \leftrightarrow \beta 2$
  - α is c or p expressing **class** or **package** coupling
  - $\beta 1$ is s or u expressing whether the left-hand side analysis is **static** or (dynamic) **unweighted**
  - $\beta 2$ is u or w expressing whether the right-hand side analysis is (dynamic) **unweighted** or (dynamic) **weighted**
- For example, the comparison of static and (dynamic) unweighted coupling metrics at class level is denoted as $c : s \leftrightarrow u$

Results

(a) Experiment 1.

	$c : s \leftrightarrow u$	$c : s \leftrightarrow w$	$c : u \leftrightarrow w$	$p : s \leftrightarrow u$	$p : s \leftrightarrow w$	$p : u \leftrightarrow w$
import	0.31+0.01	0.36+0.01	0.13-0.01	0.33	0.36-0.01	0.08
export	0.41	0.41-0.01	0.24	0.30	0.32-0.01	0.21-0.01
combined	0.35	0.41-0.01	0.29-0.01	0.29	0.33-0.01	0.23-0.01
average	0.35	0.39	0.22-0.01	0.31	0.33	0.17

(b) Experiment 2.

	$c : s \leftrightarrow u$	$c : s \leftrightarrow w$	$c : u \leftrightarrow w$	$p : s \leftrightarrow u$	$p : s \leftrightarrow w$	$p : u \leftrightarrow w$
import	0.30+0.02	0.36+0.01	0.14	0.31	0.35	0.09-0.01
export	0.41	0.43-0.01	0.26-0.01	0.30	0.33	0.22-0.01
combined	0.34+0.01	0.41-0.01	0.31-0.01	0.28	0.33-0.01	0.23
average	0.35+0.01	0.40	0.24-0.01	0.30	0.33	0.18

(c) Experiment 3.

	$c : s \leftrightarrow u$	$c : s \leftrightarrow w$	$c : u \leftrightarrow w$	$p : s \leftrightarrow u$	$p : s \leftrightarrow w$	$p : u \leftrightarrow w$
import	0.38	0.42+0.01	0.12	0.37	0.39	0.06
export	0.38	0.40	0.22	0.28	0.31	0.20
combined	0.36	0.40+0.01	0.28	0.30	0.33	0.23+0.01
average	0.37	0.41	0.21	0.32	0.35	0.17

(d) Experiment 4.

	$c : s \leftrightarrow u$	$c : s \leftrightarrow w$	$c : u \leftrightarrow w$	$p : s \leftrightarrow u$	$p : s \leftrightarrow w$	$p : u \leftrightarrow w$
import	0.37	0.42	0.12	0.36	0.39	0.06
export	0.38	0.40	0.23	0.28	0.32	0.20
combined	0.35	0.40+0.01	0.29	0.30	0.33	0.24
average	0.37	0.41	0.21	0.31	0.35	0.17

Challenges

- Data
  - Used Jira variant was unknown (Jira Core or Jira Software)
  - The encryption of package, class, and method names in the publicly available monitoring data led to different results for the Kendall-Tau distance
- Method
  - Used Kendall-Tau distance implementation was unknown
  - Included and excluded call relationships were unclear regarding lambda methods and constructor calls
- Documentation
  - Table 10 in the original paper shows the average export coupling degrees derived from weighted static dependency graphs but with the static analysis only the unweighted coupling degrees were used

Reproduction Package

Experiences in Replicating an Experiment on Comparing Static and Dynamic Coupling Metrics

The replication of the experiment conducted by Schnoor and Hasselbring Comparing Static and Dynamic Weighted Software Coupling Metrics can either be executed online or locally.

Online Replication (experiment 4)

Click on the badge to access the replication study online.

launch binder

Local Replication (experiments 1, 2, 3, 4)

1. Download and unzip or clone this repository.
2. Download the kieker monitoring logs and copy them into the following folders but keep the existing `kieker.map` files.

- Experiment 1 -> `/replication/experiment/1/traces`
- Experiment 2 -> `/replication/experiment/2/traces`
- Experiment 3 -> `/replication/experiment/3/traces`
- Experiment 4 -> `/replication/experiment/4/traces`

4. Download the Jira byte code and extract the archives into the folowing folders.

- Jira byte code v7.3.0 -> `/replication/experiment/1/system`
- Jira byte code v7.4.3 -> `/replication/experiment/2/system`
- Jira byte code v7.7.1 -> `/replication/experiment/3/system`
- Jira byte code v7.7.1 -> `/replication/experiment/4/system`

6. Execute the batch files `1-scan.bat` and `2-aggregate-calls.bat` for each experiment one after another.
7. Execute `3-start-neo4j-server.bat` and explore the created dependency graphs of the selected experiment at http://localhost:7474/.
8. Before you can execute the jupyter notebook `/jupyter/Replication of Comparing Coupling Metrics.ipynb`, you have to set up a corresponding environment. You can for example download Anaconda and install it. Furthermore, make sure that `jupyter notebook`, `pandas`, and `py2neo` are installed in the environment.

[https://github.com/softvis-research/coupling-metrics-replication]

Conclusion

- Successfully replicated the original experiment
- Provide a reproduction package
- Researchers reproducing or replicating an empirical study should contact the authors of the original study as early as possible
- Researchers wanting to ensure reproducibility and replicability of their empirical study should check
  - if the empirical data, the applied methods, and the obtained results are sufficiently described in the paper and/or documented elsewhere
  - if data encryption affects the reproduction or replication of the results

References

- ACM Terminology Version 1.0, 2020. Available at: https://www.acm.org/publications/policies/artifact-review-badging
- ACM Terminology Version 1.1, 2020. Available at: https://www.acm.org/publications/policies/artifact-review-and-badging-current
- Briand, L., Emam, K. El & Morasca, S., 1996. On the application of measurement theory in software engineering. Empirical Software Engineering, 1(1), Pp. 61–88. Available at: https://doi.org/10.1007/BF00125812.
- Müller, R., Mahler, D., Hunger, M., Nerche, J., Harrer, M., 2018. Towards an Open Source Stack to Create a Unified Data Source for Software Analysis and Visualization. In *6th IEEE Working Conference on Software Visualization*. Madrid, Spain: IEEE.
- Müller, R. & Fischer, M., 2019. Graph-Based Analysis and Visualization of Software Traces. In 10th Symposium on Software Performance. Würzburg, Germany.
- Müller, R. & Strempel, T., 2020. Graph-Based Performance Analysis at System- and Application-Level. In 11th Symposium on Software Performance. Leipzig, Germany.
- Schnoor, H. & Hasselbring, W., 2020. Comparing Static and Dynamic Weighted Software Coupling Metrics. Computers, 9(2), P. 24. Available at: https://www.mdpi.com/2073-431X/9/2/24.

Your Questions

?

mail@rmllr.de

https://twitter.com/rimllr

https://github.com/rmllr

https://rmllr.de

Reproduction vs. Replication

Artifact Review and Badging Version 1.0 until August 24, 2020

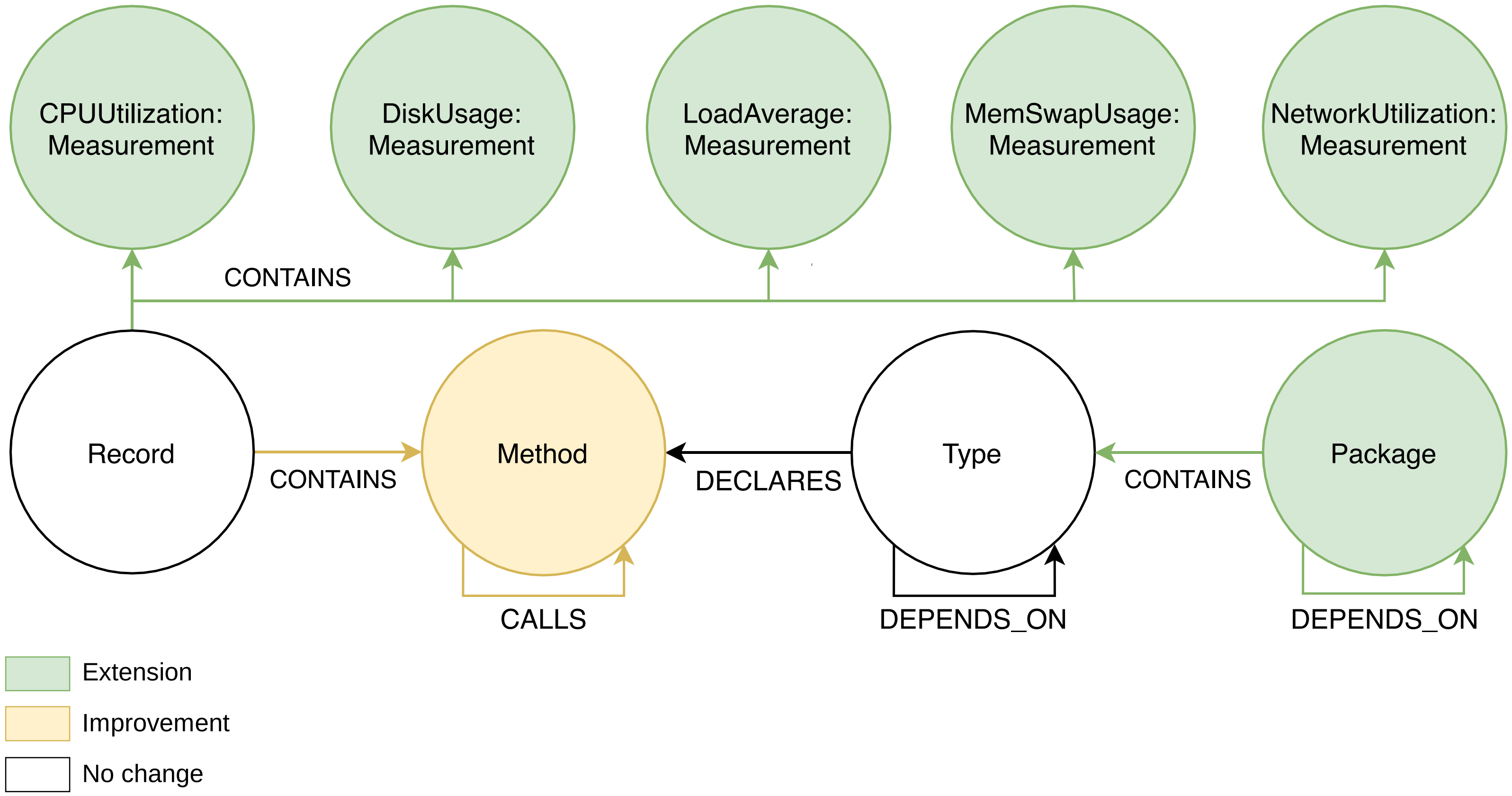
- **Reproducibility (Different team, different experimental setup)***
  - The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author's own artifacts.
- **Replicability (Different team, same experimental setup)***
  - The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.

Artifact Review and Badging Version 1.1 since August 24, 2020

- **Reproducibility (Different team, same experimental setup)***
  - The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author’s own artifacts.
- **Replicability (Different team, different experimental setup)***
  - The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.

[ACM Terminology Version 1.0 and 1.1 2020]

Revised Kieker Graph Schema



[Müller & Strempel 2020]