Performance Monitoring of Database Operations

Christian Zirkelbach

July 29, 2015

Outline

- Performance of new or legacy software systems is insufficient
- Performance problems or bottlenecks are supposed or detected
- Often related to database operations

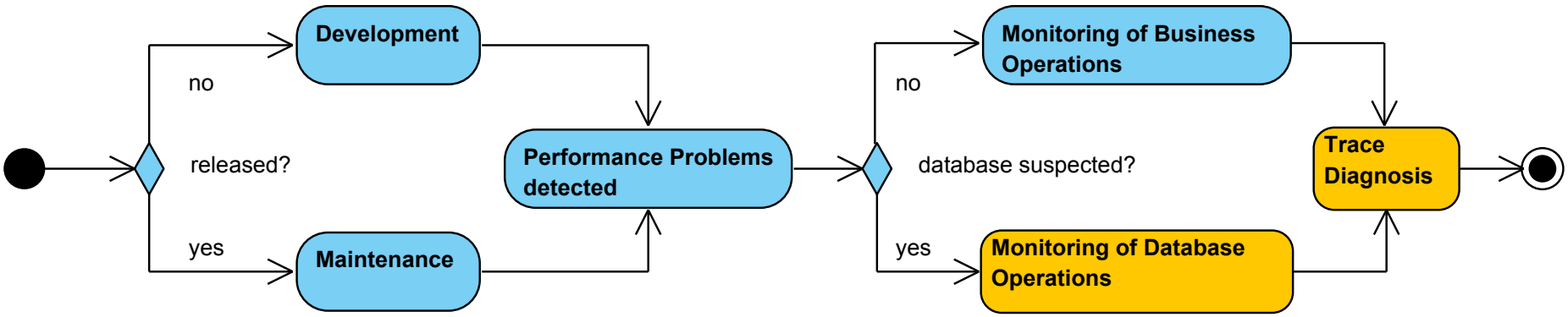

Figure: Enriched performance issue detection workflow

- Monitoring Component
  - executed SQL and prepared SQL statements
  - their call parameters
  - execution times
- Analysis & Visualization Component
  - analyzing recorded data
  - filtering, sorting, ...
  - grouping prepared statements

Goals

1. Identification of Performance Analysis Methods and Tools
2. Implementation of a Tool for Database Performance Analysis
3. Generic Monitoring Approach
4. Evaluation of the Developed Tool

Approach

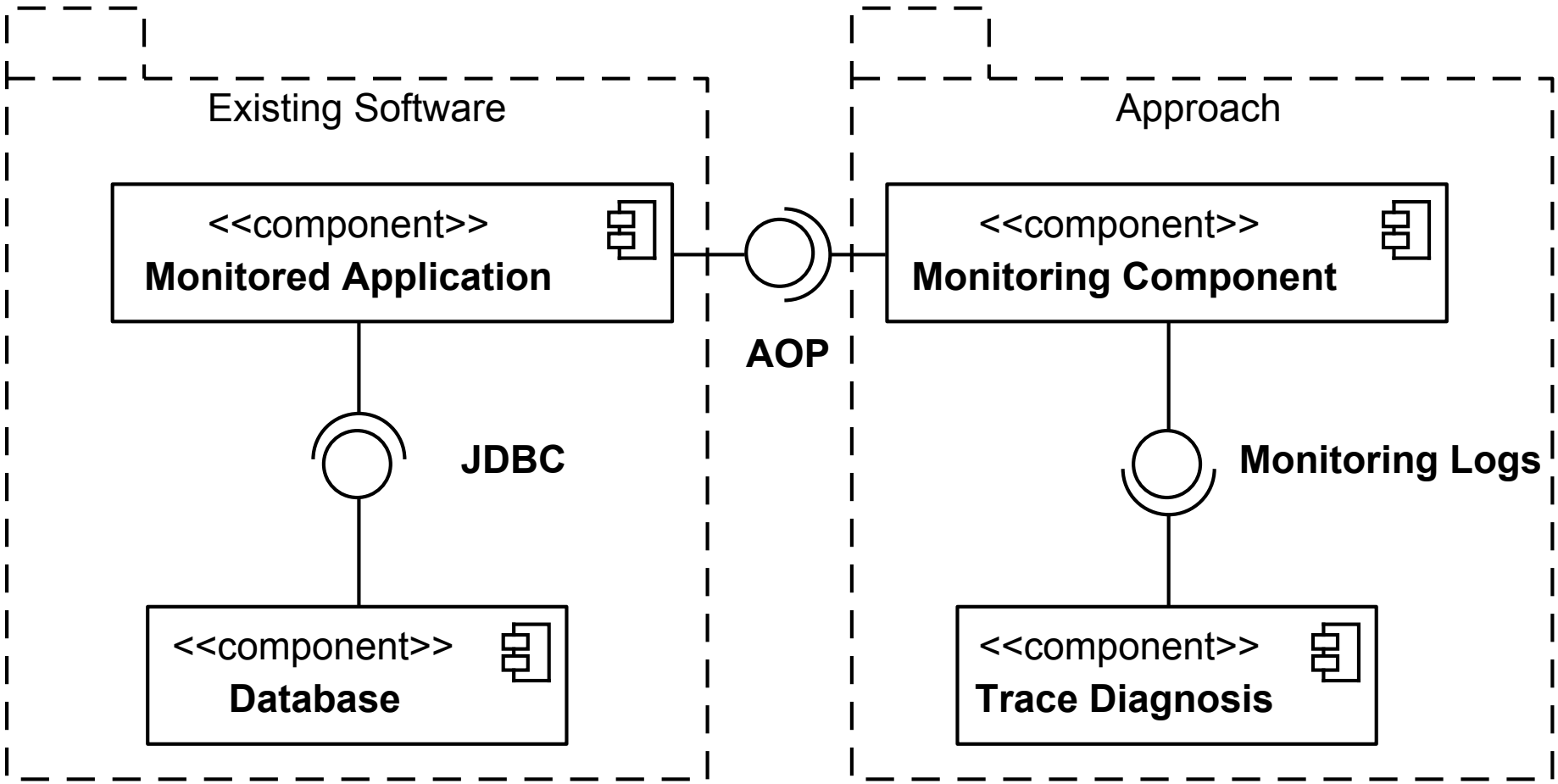

Figure: Architecture of our software system as component diagram

SQL relationships

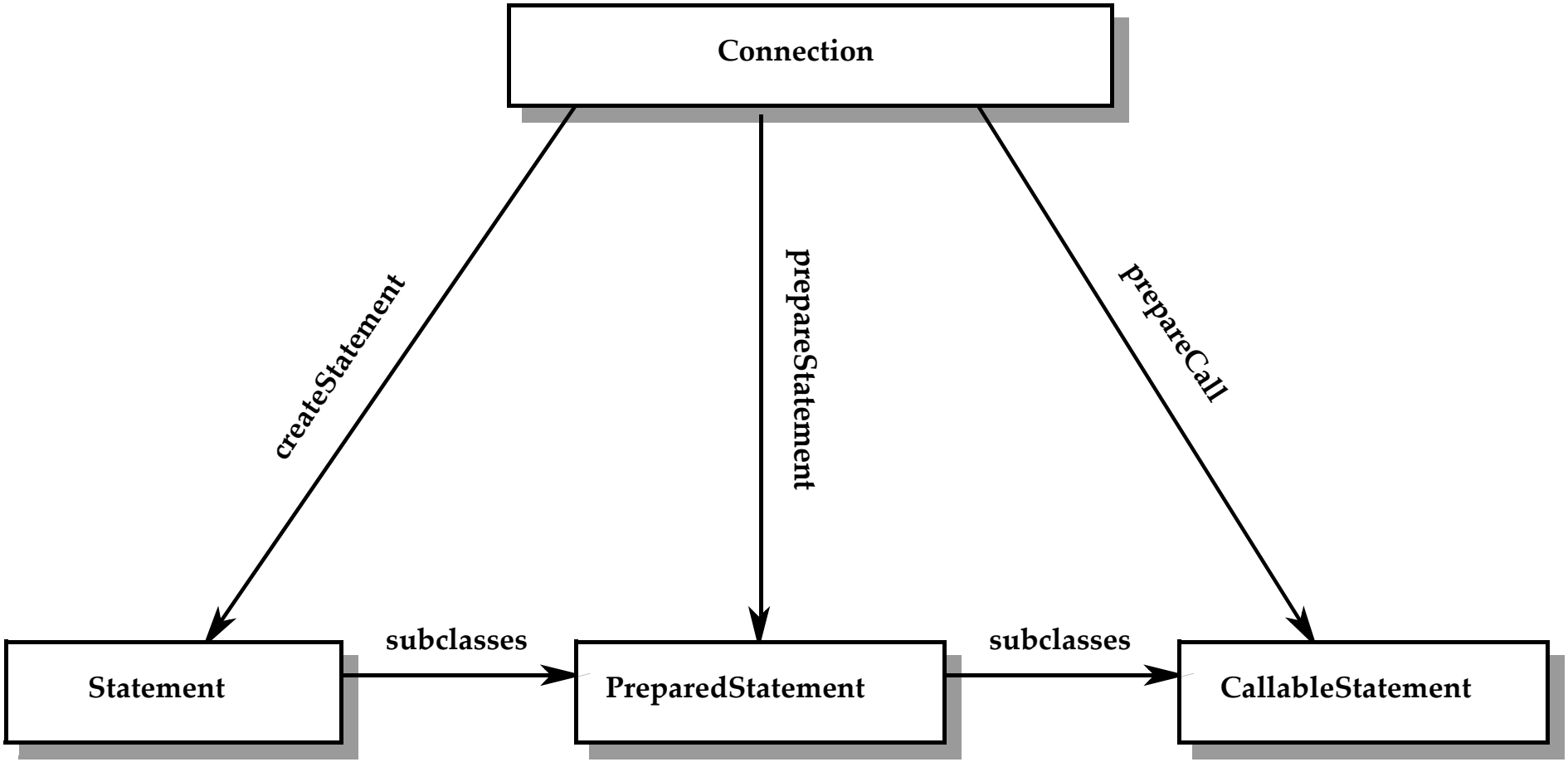

Figure: Relationships between major classes and interfaces in *java.sql*

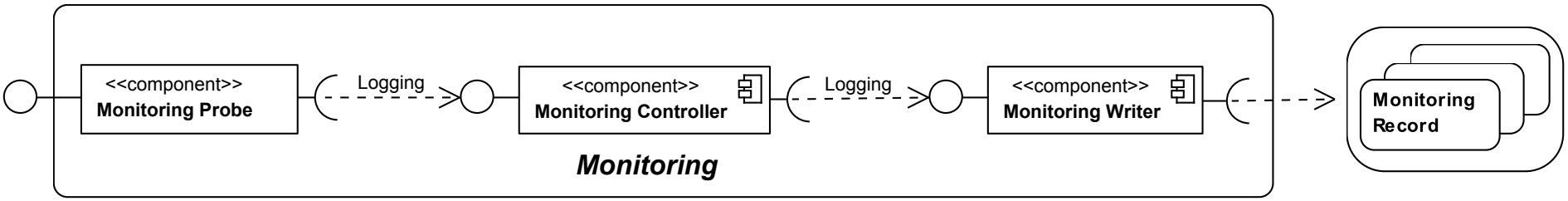

Figure: Monitoring components

- record type: differs between before or after event
- timestamp: represents date and time of the record
- operation name: full Java class name
- return type: e.g., ResultSet, boolean or int
- return value: e.g., number of affected database records
- operation arguments: e.g., SQL statement

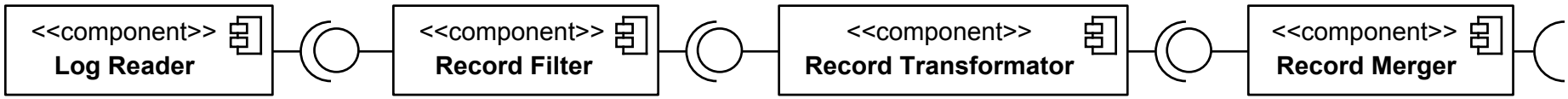

Figure: Generic Monitoring Record Processing based on P&F architecture

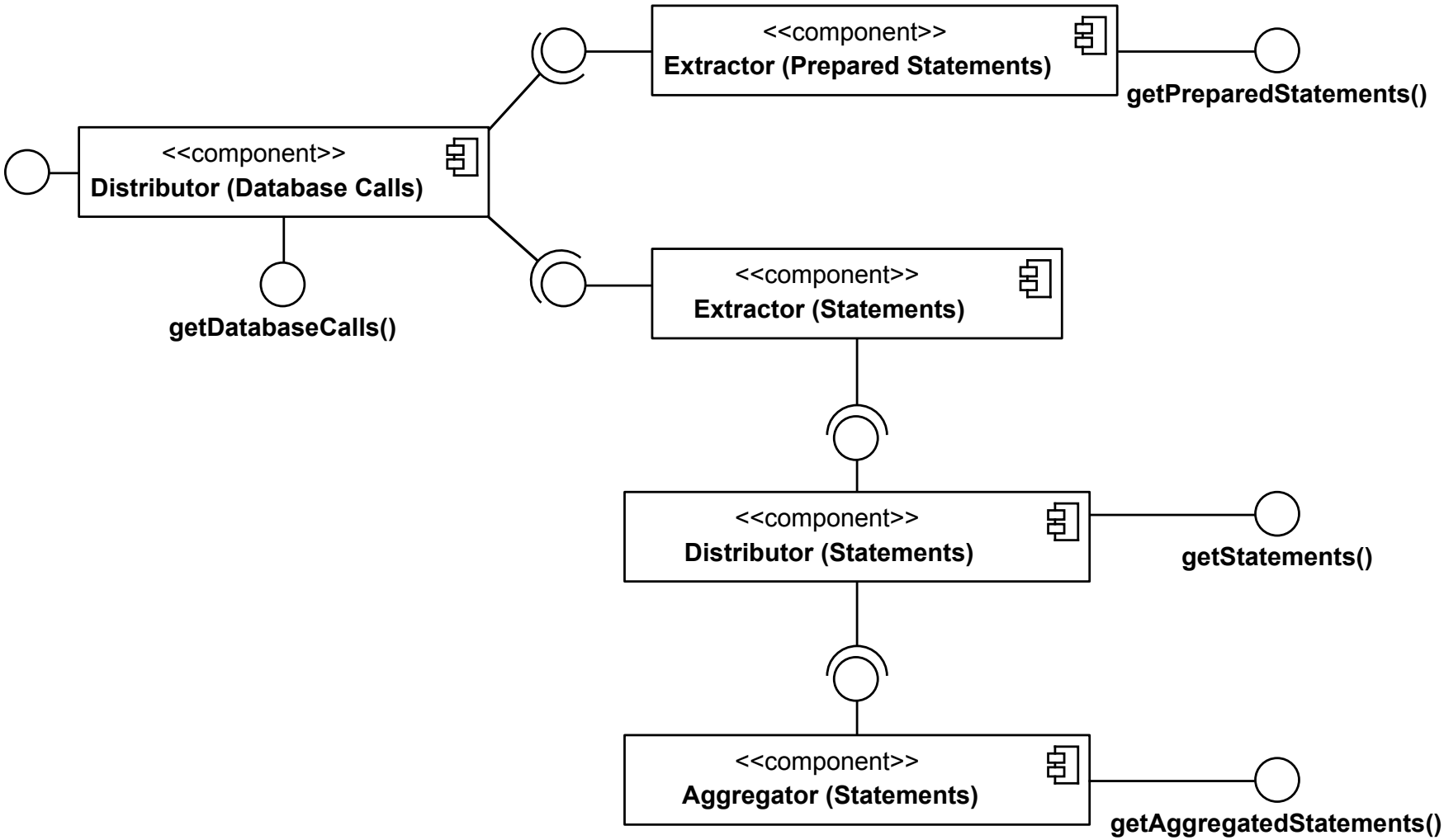

Figure: Specific Database Call Handling based on P&F architecture

Mock-up: Prepared Statements

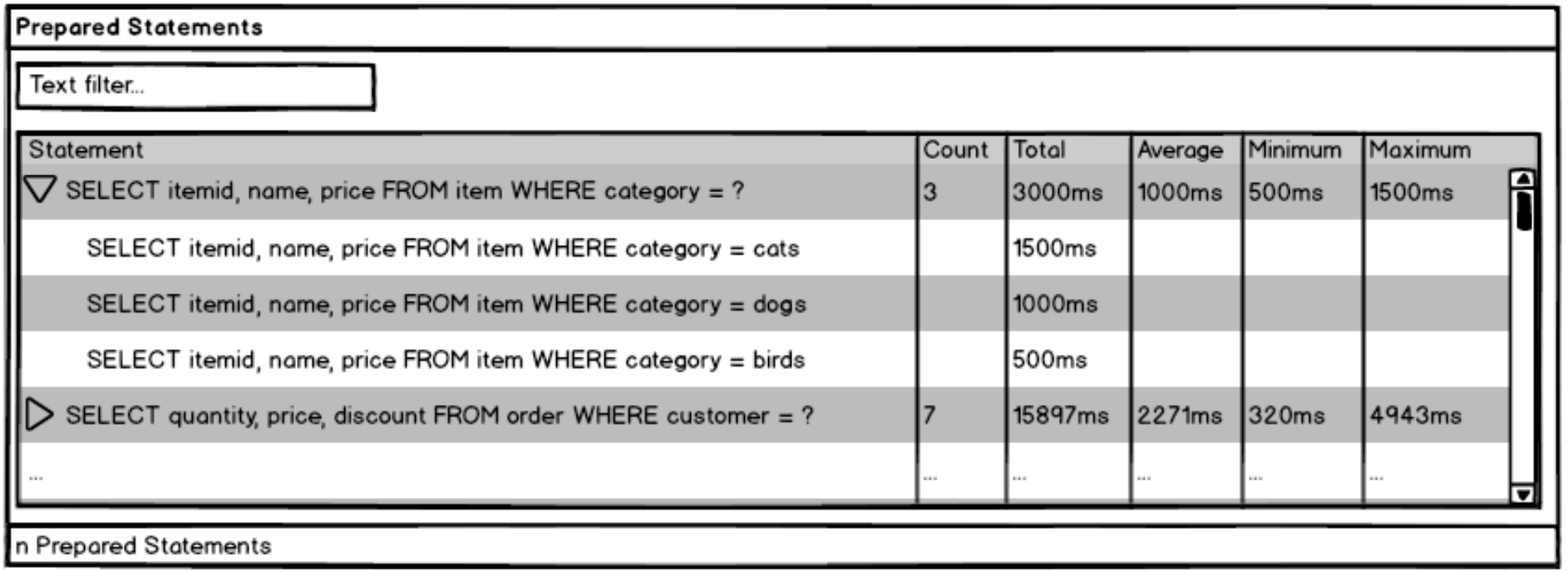

Figure: Mock-up view based on *Call Tree Views* [De Pauw et al. (2002)]

Implementation

Statements View

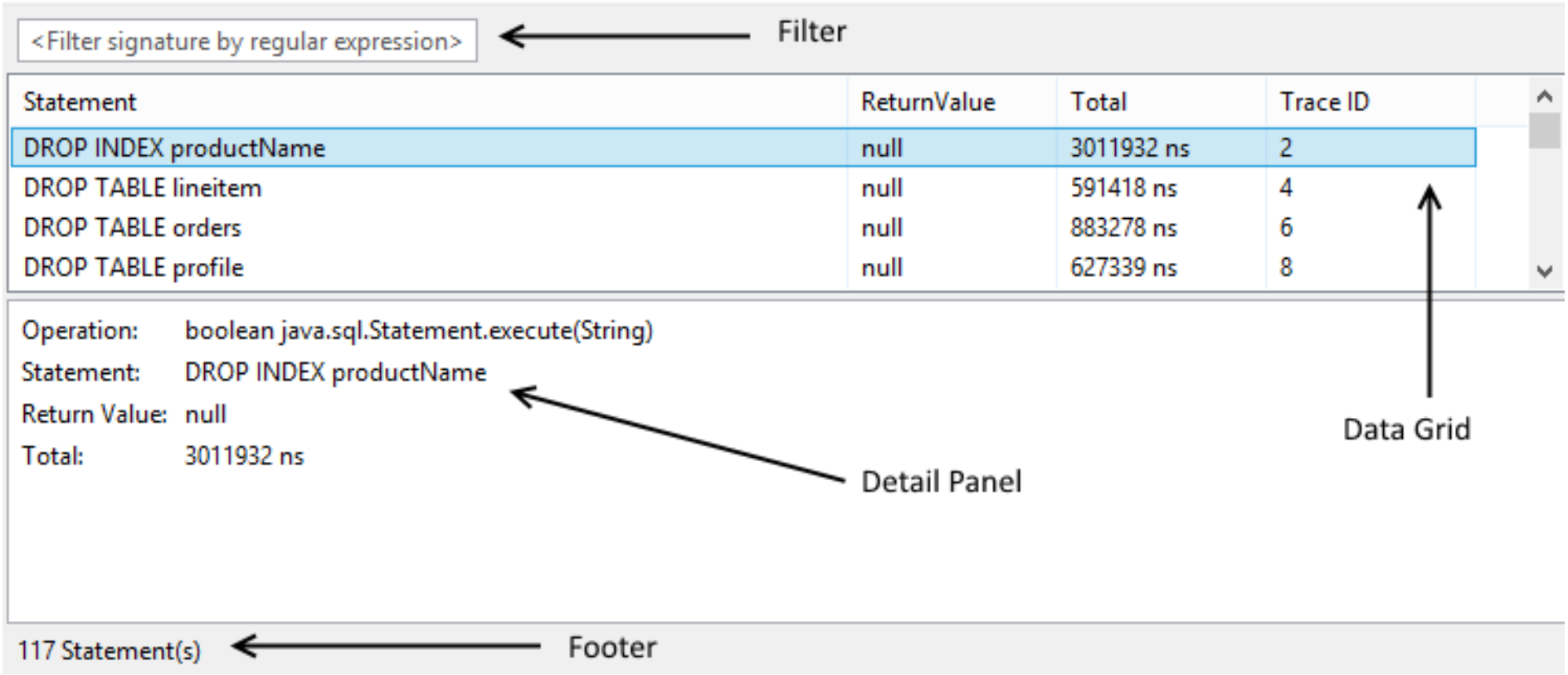

Figure: Statement view screenshot

Aggregated Statements View

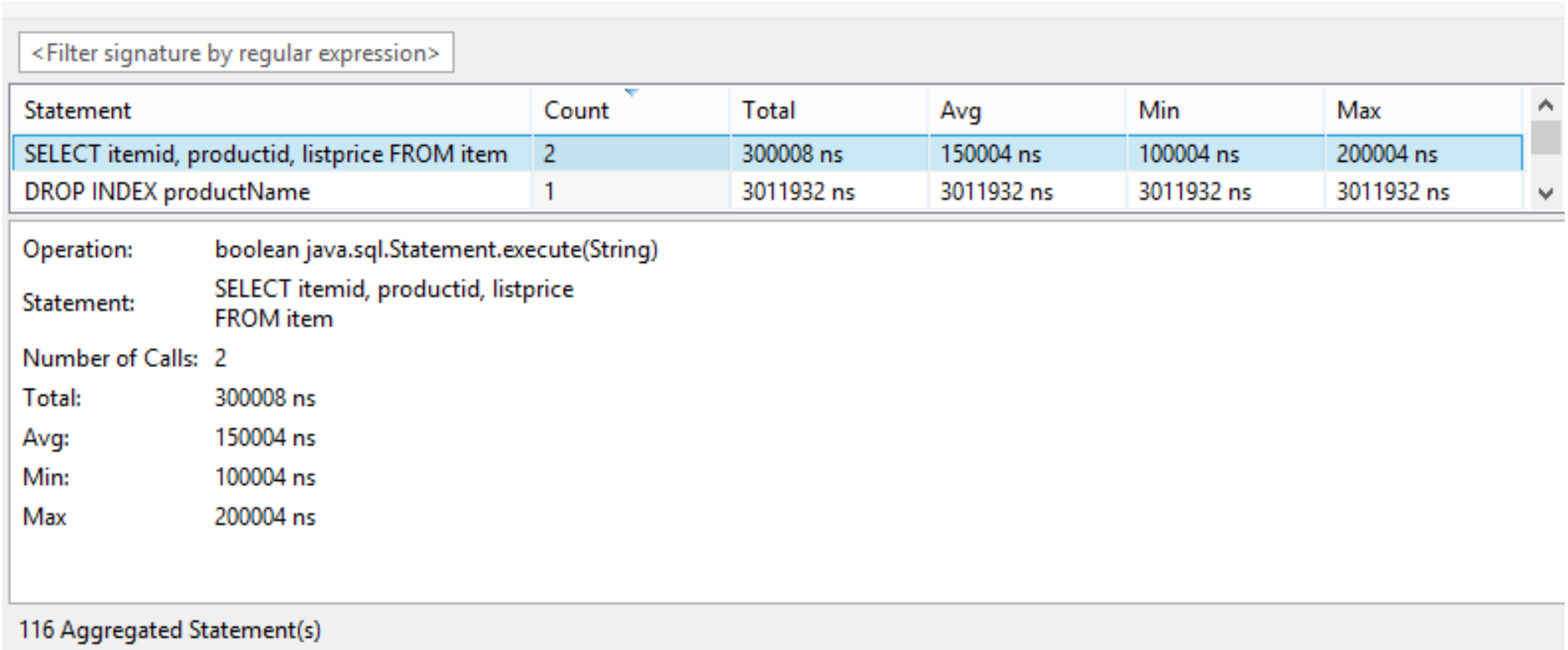

Figure: Aggregated statement view screenshot

Prepared Statements View

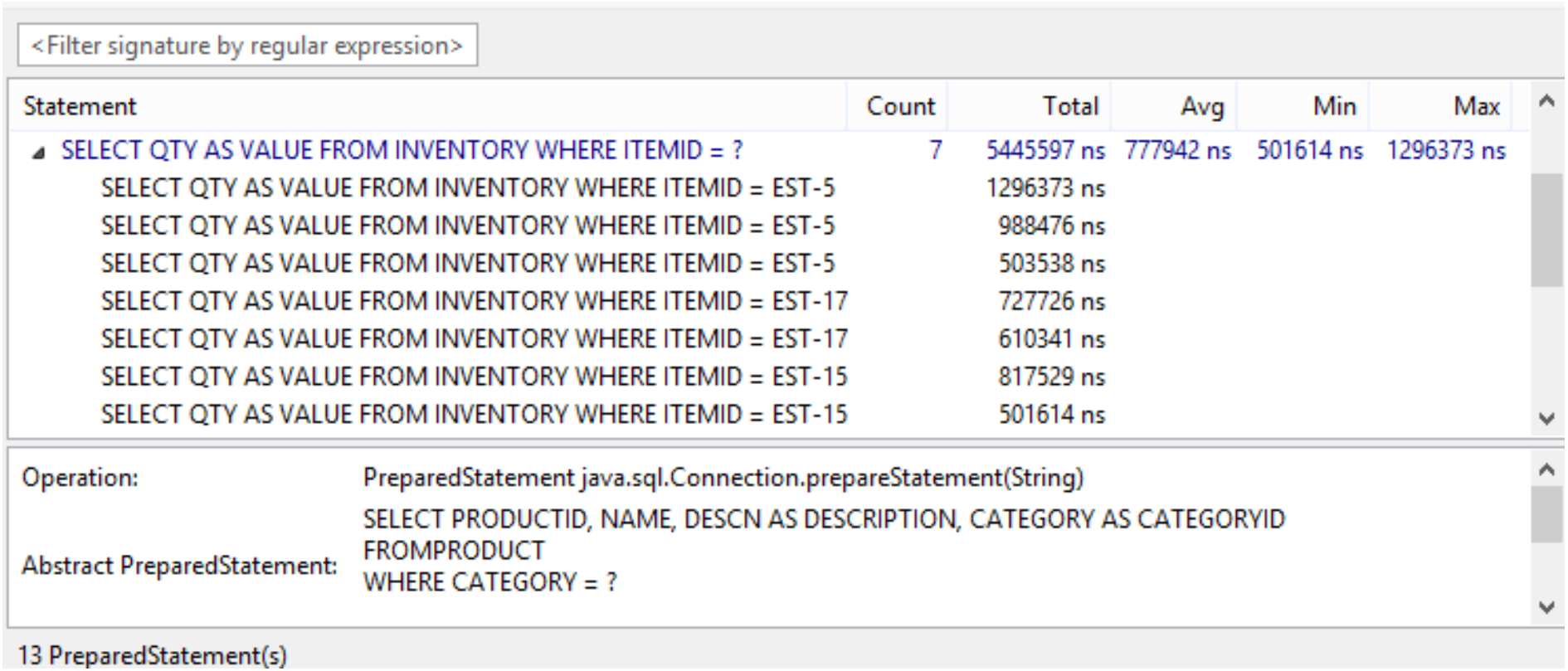

Figure: Prepared statement view screenshot

Evaluation

- Usability Test Experiment
- Based on a questionnaire, 36 participants
- Questionnaires have been used a long time to evaluate user interfaces [Root and Draper(1983)]

3 Statements

3.1 Name the Trace ID and response time (in ms) of the statement, that has the highest response time.

3.2 What is it's underlying calling Java operation?

3.3 What kind of SQL statement took the lowest amount of time?

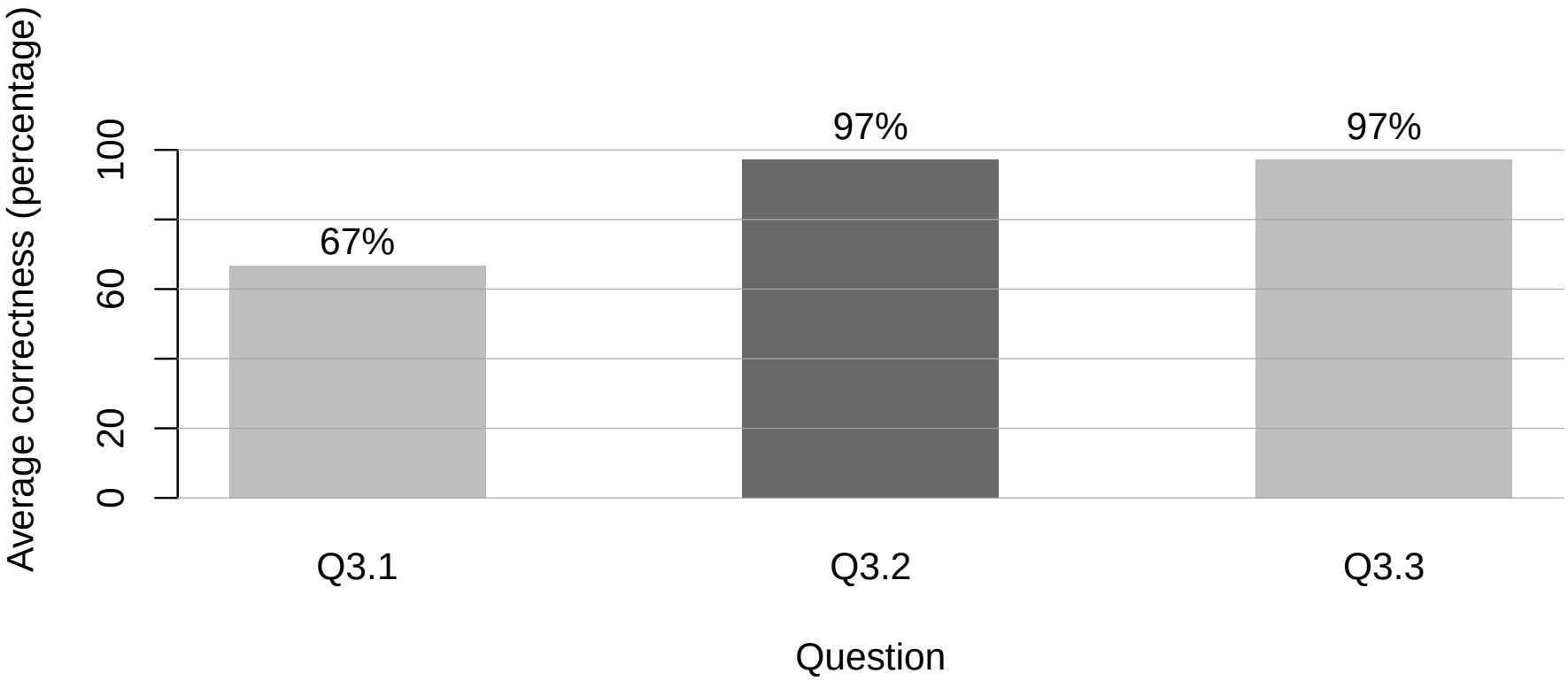

Figure: Average correctness per question within the statements part

6 Debriefing Questions

6.1 How difficult was it to **navigate** through the program?

very difficult ☐ ☐ ☐ ☐ ☐ very easy

6.2 How difficult was it to **filter** and **sort** database statements for specific problems?

very bad ☐ ☐ ☐ ☐ ☐ very good

6.3 Was the program **easy to use?**

very difficult ☐ ☐ ☐ ☐ ☐ very easy

6.4 How was your **overall impression** of the tool?

very bad ☐ ☐ ☐ ☐ ☐ very good

Debriefing Questions

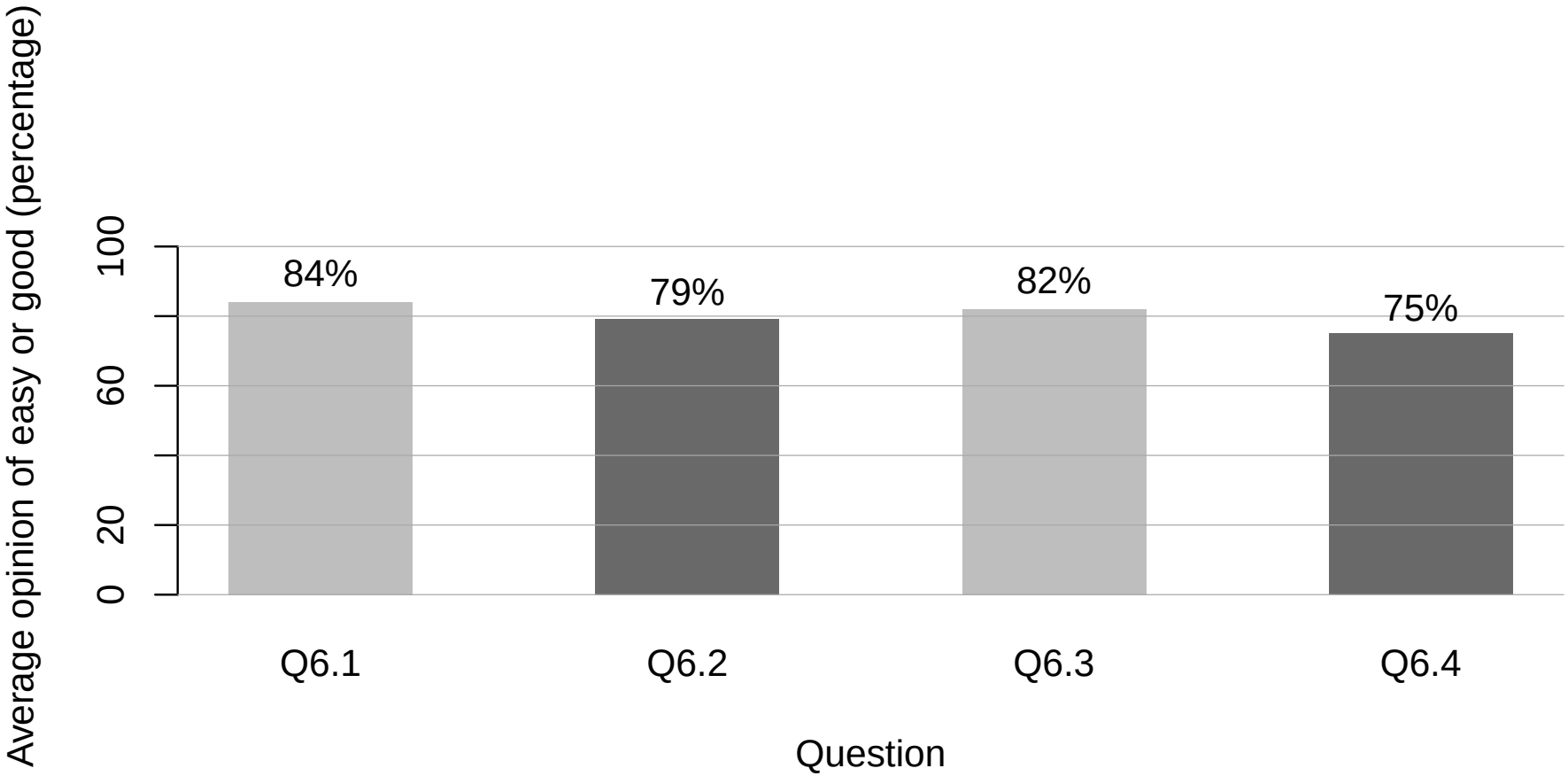

Figure: Average rating of easy or good per question within the debriefing questions

Summary

- Positive feedback from participants
- High average correctness rates
- Positive usability rating

- Cloud Monitoring [Ma et al. (2012)]

- Software supports performance analysis on database operations
- Monitoring component versatile (Kieker Trace Diagnosis and ExplorViz [Fittkau 2015])
- Approach validated through conducted experiment

- Easier filtering, e.g., based on substrings
- More visualization options, e.g., 3D visualizations
- Further experiments, e.g., controlled experiments similar to [Fittkau et al., May 2015a]

References I

AspectJ language extension.

URL `http://www.eclipse.org/aspectj/`.

G. Antoniol, M. Di Penta, and M. Zazzara.

Understanding Web applications through dynamic analysis.
In *Program Comprehension, 2004. Proceedings. 12th IEEE International Workshop on*, pages 120–129, June 2004.

C. Artho and A. Biere.

Combined Static and Dynamic Analysis.
In *Proceedings of the 1st International Workshop on Abstract Interpretation of Object-oriented Languages (AIOOL 2005)*, volume 131, pages 3 – 14, 2005.

C. Artho, V. Schuppan, A. Biere, P. Eugster, M. Baur, and B. Zweimüller.

JNuke: Efficient Dynamic Analysis for Java.
In R. Alur and D. Peled, editors, *Computer Aided Verification*, volume 3114 of *Lecture Notes in Computer Science*, pages 462–465. Springer Berlin Heidelberg, 2004.

T. Ball.

The Concept of Dynamic Analysis.
In O. Nierstrasz and M. Lemoine, editors, *Software Engineering*, volume 1687 of *Lecture Notes in Computer Science*, pages 216–234. Springer Berlin Heidelberg, 1999.

D. Balzarotti, M. Cova, V. Felmetsger, N. Jovanovic, E. Kirda, C. Kruegel, and G. Vigna.

Saner: Composing Static and Dynamic Analysis to Validate Sanitization in Web Applications.
In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 387–401, May 2008.

References II

References

M. Bastian, S. Heymann, M. Jacomy, et al.

Gephi: an open source software for exploring and manipulating networks.
In *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media*, pages 361–362, 2009.

S. Becker, W. Hasselbring, A. van Hoorn, S. Kounev, and R. Reussner.

Proceedings of the 2014 Symposium on Software Performance (SOSP'14): Joint Descartes/Kieker/Palladio Days.
2014.

G. Canfora Harman and M. Di Penta.

New Frontiers of Reverse Engineering.
In *2007 Future of Software Engineering*, FOSE '07, pages 326–341, Washington, DC, USA, 2007. IEEE Computer Society.

A. Chawla and A. Orso.

A Generic Instrumentation Framework for Collecting Dynamic Information.
SIGSOFT Softw. Eng. Notes, 29(5):1–4, Sept. 2004.

S. Chiba.

Javassist - a reflection-based programming wizard for Java.
In *Proceedings of OOPSLA'98 Workshop on Reflective Programming in C++ and Java*, page 174, 1998.

E. Chikofsky and I. Cross, J.H.

Reverse engineering and design recovery: a taxonomy.
Software, IEEE, 7(1):13–17, Jan 1990.

J. M. Coble, J. Karat, M. J. Orland, and M. G. Kahn.

Iterative usability testing: ensuring a usable clinical workstation.
In *Proceedings of the AMIA Annual Fall Symposium*, page 744. American Medical Informatics Association, 1997.

References III

B. Cornelissen, A. Zaidman, A. van Deursen, L. Moonen, and R. Koschke.

A Systematic Survey of Program Comprehension through Dynamic Analysis.
Software Engineering, IEEE Transactions on, 35(5):684–702, Sept 2009.

W. De Pauw, E. Jensen, N. Mitchell, G. Sevitsky, J. Vlissides, and J. Yang.

Visualizing the execution of Java programs.
In *Software Visualization*, pages 151–162. Springer, 2002.

S. Ducasse and D. Pollet.

Software Architecture Reconstruction: A Process-Oriented Taxonomy.
Software Engineering, IEEE Transactions on, 35(4):573–591, July 2009.

J. Ehlers, A. van Hoorn, J. Waller, and W. Hasselbring.

Self-adaptive Software System Monitoring for Performance Anomaly Localization.
In *Proceedings of the 8th ACM International Conference on Autonomic Computing*, ICAC '11, pages 197–200, New York, NY, USA, 2011. ACM.

H. Eichelberger and K. Schmid.

Flexible resource monitoring of Java programs.
Journal of Systems and Software, 93(0):163 – 186, 2014.

T. Eisenbarth, R. Koschke, and D. Simon.

Aiding program comprehension by static and dynamic feature analysis.
In *Software Maintenance, 2001. Proceedings. IEEE International Conference on*, pages 602–611, 2001.

M. D. Ernst.

Static and dynamic analysis: Synergy and duality.
In *WODA 2003: ICSE Workshop on Dynamic Analysis*, pages 24–27, Portland, OR, May 9, 2003.

References IV

References

F. Fittkau.

Live Trace Visualization for System and Program Comprehension in Large Software Landscapes.
In *PhD Topic Presentation*, Juli 2015.

F. Fittkau, J. Waller, P. C. Brauer, and W. Hasselbring.

Scalable and Live Trace Processing with Kieker Utilizing Cloud Computing.
In *Proceedings of the Symposium on Software Performance: Joint Kieker/Palladio Days 2013*, volume 1083, pages 89–98. CEUR Workshop Proceedings, November 2013a.

F. Fittkau, J. Waller, C. Wulf, and W. Hasselbring.

Live Trace Visualization for Comprehending Large Software Landscapes: The ExplorViz Approach.
In *1st IEEE International Working Conference on Software Visualization (VISSOFT 2013)*, pages 1–4, September 2013b.

F. Fittkau, S. Finke, W. Hasselbring, and J. Waller.

Comparing Trace Visualizations for Program Comprehension through Controlled Experiments.
In *23rd IEEE International Conference on Program Comprehension (ICPC 2015)*, Mai 2015a.

F. Fittkau, S. Roth, and W. Hasselbring.

ExplorViz: Visual Runtime Behavior Analysis of Enterprise Application Landscapes.
In *23rd European Conference on Information Systems (ECIS 2015)*, Mai 2015b.

Florian Fittkau.

ExplorViz Project, 2015.
URL `http://www.explorviz.net/`.

M. Fowler.

Patterns of enterprise application architecture.
Addison-Wesley Longman Publishing Co., Inc., 2002.

L. Froihofer, G. Glos, J. Osrael, and K. M. Goeschka.

Overview and evaluation of constraint validation approaches in Java.
In *Proceedings of the 29th international conference on Software Engineering*, pages 313–322. IEEE Computer Society, 2007.

E. Gamma, R. Helm, R. Johnson, and J. Vlissides.

Design patterns: elements of reusable object-oriented software.
Pearson Education, 1994.

G. Hamilton, R. Cattell, M. Fisher, et al.

JDBC Database Access with Java, volume 7.
Addison Wesley, 1997.

W. Hasselbring.

Reverse Engineering of Dependency Graphs via Dynamic Analysis.
In *Proceedings of the 5th European Conference on Software Architecture: Companion Volume*, ECSA '11, pages 5:1–5:2, New York, NY, USA, 2011. ACM.

R. Jung, R. Heinrich, and E. Schmieders.

Model-driven Instrumentation with Kieker and Palladio to forecast Dynamic Applications.
In *Proceedings Symposium on Software Performance: Joint Kieker/Palladio Days 2013 (KPDAYS 2013)*, volume 1083 of *CEUR Workshop Proceedings*, pages 99–108. CEUR, November 2013.

J. Karat.

Evolving the scope of user-centered design.
Communications of the ACM, 40(7):33–38, 1997.

References VI

G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin.

Aspect-oriented programming.
In M. Akşit and S. Matsuoka, editors, *ECOOP'97 — Object-Oriented Programming*, volume 1241 of *Lecture Notes in Computer Science*, pages 220–242. Springer Berlin Heidelberg, 1997.

G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. Griswold.

An Overview of AspectJ.
In J. Knudsen, editor, *ECOOP 2001 — Object-Oriented Programming*, volume 2072 of *Lecture Notes in Computer Science*, pages 327–354. Springer Berlin Heidelberg, 2001.

C. Knight and M. Munro.

Virtual but visible software.
In *Proceedings of IEEE International Conference on Information Visualization*, pages 198–205, 2000.

R. Likert.

A technique for the measurement of attitudes.
Archives of psychology, 1932.

A. M. Lund.

Measuring usability with the USE questionnaire.
Usability interface, 8(2):3–6, 2001.

K. Ma, R. Sun, and A. Abraham.

Toward a lightweight framework for monitoring public clouds.
In *Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on*, pages 361–365. IEEE, 2012.

H. A. Müller, J. H. Jahnke, D. B. Smith, M.-A. Storey, S. R. Tilley, and K. Wong.

Reverse Engineering: A Roadmap.
In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 47–60, New York, NY, USA, 2000. ACM.

J. Neilson, C. Woodside, D. Petriu, and S. Majumdar.

Software bottlenecking in client-server systems and rendezvous networks.
Software Engineering, IEEE Transactions on, 21(9):776–782, Sep 1995.

J. Nielsen and T. K. Landauer.

A mathematical model of the finding of usability problems.
In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 206–213. ACM, 1993.

Oracle.

JDBC™ 4.1 Specification, 2011.

R. Pooley.

Software Engineering and Performance: A Roadmap.
In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 189–199, New York, NY, USA, 2000. ACM.

K. Project.

Kieker User Guide, Apr. 2013.
URL `http://kieker-monitoring.net/documentation/`.

References VIII

References

M. Rohr, A. van Hoorn, S. Giesecke, J. Matevska, W. Hasselbring, and S. Alekseev.

Trace-Context Sensitive Performance Profiling for Enterprise Software Applications.
In S. Kounev, I. Gorton, and K. Sachs, editors, *Performance Evaluation: Metrics, Models and Benchmarks*, volume 5119 of *Lecture Notes in Computer Science*, pages 283–302. Springer Berlin Heidelberg, 2008.

R. W. Root and S. Draper.

Questionnaires as a software evaluation tool.
In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 83–87. ACM, 1983.

A. Sabetta and H. Koziolek.

Measuring Performance Metrics: Techniques and Tools.
In I. Eusgeld, F. Freiling, and R. Reussner, editors, *Dependability Metrics*, volume 4909 of *Lecture Notes in Computer Science*, pages 226–232. Springer Berlin Heidelberg, 2008.

J. Sauro and J. R. Lewis.

Quantifying the user experience: Practical statistics for user research.
Elsevier, 2012.

E. Stroulia and T. Systä.

Dynamic Analysis for Reverse Engineering and Program Understanding.
SIGAPP Appl. Comput. Rev., 10(1):8–17, Apr. 2002.

L. Titchkosky, M. Arlitt, and C. Williamson.

A Performance Comparison of Dynamic Web Technologies.
SIGMETRICS Perform. Eval. Rev., 31(3):2–11, Dec. 2003.
ISSN 0163-5999.

References IX

P. Tonella and M. Ceccato.

Aspect mining through the formal concept analysis of execution traces.
In *Reverse Engineering, 2004. Proceedings. 11th Working Conference on*, pages 112–121, Nov 2004.

T. Tullis and W. Albert.

Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

T. S. Tullis and J. N. Stetson.

A comparison of questionnaires for assessing website usability.
In *Usability Professional Association Conference*, pages 1–12, 2004.

A. van Hoorn, M. Rohr, W. Hasselbring, J. Waller, J. Ehlers, S. Frey, and D. Kieselhorst.

Continuous Monitoring of Software Services: Design and Application of the Kieker Framework.
Technical Report TR-0921, Department of Computer Science, Kiel University, Germany, Nov. 2009.

A. van Hoorn, S. Frey, W. Goerigk, W. Hasselbring, H. Knoche, S. Köster, H. Krause, M. Porembski, T. Stahl, M. Steinkamp, and N. Wittmüss.

DynaMod Project: Dynamic Analysis for Model-Driven Software Modernization .
In *Joint Proceedings of the 1st International Workshop on Model-Driven Software Migration (MDSM 2011) and the 5th International Workshop on Software Quality and Maintainability (SQM 2011)* , volume 708 of *CEUR Workshop Proceedings*, pages 12–13, 2011a.
Invited paper.

A. van Hoorn, H. Knoche, W. Goerigk, and W. Hasselbring.

Model-Driven Instrumentation for Dynamic Analysis of Legacy Software Systems .
In *Proceedings of the 13. Workshop Software-Reengineering (WSR 2011)*, pages 26–27, Bad Honnef, Germany, May 2-4, 2011, May 2011b.

A. van Hoorn, J. Waller, and W. Hasselbring.

Kieker: A Framework for Application Performance Monitoring and Dynamic Software Analysis.
In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering (ICPE 2012)*, pages 247–248. ACM, Apr. 2012a.

A. van Hoorn, J. Waller, and W. Hasselbring.

Kieker: A Framework for Application Performance Monitoring and Dynamic Software Analysis.
In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, ICPE '12, pages 247–248, New York, NY, USA, 2012b. ACM.

A. Wert.

Performance Problem Diagnostics by Systematic Experimentation.
In *Proceedings of the 18th International Doctoral Symposium on Components and Architecture*, WCOP '13, pages 1–6, New York, NY, USA, 2013. ACM.

R. Wettel and M. Lanza.

CodeCity: 3D Visualization of Large-Scale Software.
In *Companion of the 30th international conference on Software engineering*, pages 921–922. ACM, 2008.

M. Woodside, G. Franks, and D. Petriu.

The Future of Software Performance Engineering.
In *Future of Software Engineering, 2007. FOSE '07*, pages 171–187, May 2007.

C. Wulf, N. C. Ehmke, and W. Hasselbring.

Toward a Generic and Concurrency-Aware Pipes & Filters Framework.
In *Symposium on Software Performance 2014: Joint Descartes/Kieker/Palladio Days*, November 2014.

References XI

G. Xu, N. Mitchell, M. Arnold, A. Rountev, and G. Sevitsky.

Software Bloat Analysis: Finding, Removing, and Preventing Performance Problems in Modern Large-scale Object-oriented Applications.

In *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, FoSER '10, pages 421–426, New York, NY, USA, 2010. ACM.