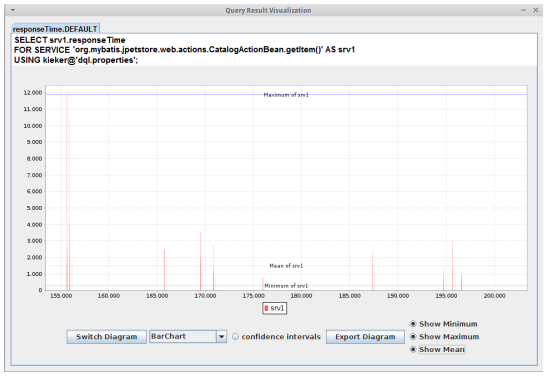

Figure 5: Visualization of performance analysis results

visualization, or no visualization at all. Our declarative approach allows for generalization of performance analysis visualizations, and reuse of these visualizations for multiple performance evaluation techniques.

PAVO framework [10] provides result visualization for a given performance analysis result. It includes several features such as a decomposition into multiple diagrams, flexibility in changing diagram types, and enrichment using aggregated metrics. To illustrate, Figure 5 shows a visualization for measurements triggered by DQL.

6 AUTOMATED PERFORMANCE MODEL EXTRACTION

We see a reluctance from industry to adopt model-based analysis approaches due to the required expertise and modeling effort. Building models from scratch in an editor does not scale for medium and large-scale systems available in industry. In order to automatically derive performance models we propose the use of Performance Model Extractor (PMX) [12], for which we also provide a web service [9]. PMX extracts architectural performance models from application performance management (APM) data.

The parametrization of the required resource demands in order to complete the DML model can be done via Library for Resource Demand Estimation (LibReDE). LibReDE is a library of ready-to-use implementations of state-of-the-art approaches for resource demand estimation, which can be used for online and offline analysis [7]. It provides eight statistical estimation approaches to derive the resource demands based on generic system- and application-level measurements.

7 MODELING OF PARAMETRIC DEPENDENCIES

Currently, automated model extraction cannot recognize when model parameters depend on a range of input characteristics or various system parametrizations. In order to reflect parametric dependencies within a performance model, explicit modeling of input parameters and description of resource demands as a function of input parameters has been shown to be effective. DML provides so-called relationships to model dependencies between various parameters and therefore predict the impact of workload changes on model parameters. Additionally, correlation relationships allow to estimate parameters that cannot be monitored, by defining correlations to measurable variables.

8 CONCLUSION

This tutorial paper addresses tools for automation of software performance engineering approaches. We present a declarative language to specify performance concerns, as well as demonstrate tools which can be used to automatically answer them based on measurements and software performance models, and a framework to provide decision support. To enable model-based analysis, we discuss tools for the efficient creation of performance models.

ACKNOWLEDGMENTS

This work is supported by the German Research Foundation (DFG) in the Priority Programme "DFG-SPP 1593: Design For Future—Managed Software Evolution" (HO 5721/1-1 and KO 3445/15-1).

REFERENCES

[1] Matthias Blohm, Maksim Pahlberg, Sebastian Vogel, Jürgen Walter, and Dusan Okanovic. 2016. Kieker4DQL: Declarative Performance Measurement. In *Proceedings of the 2016 Symposium on Software Performance (SSP)*.

[2] Fabian Gorsler, Fabian Brosig, and Samuel Kounev. 2014. Performance Queries for Architecture-Level Performance Models. In *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering (ICPE 2014)*. ACM, New York, USA, 99–110.

[3] Nikolaus Huber, Fabian Brosig, Simon Spinner, Samuel Kounev, and Manuel Bähr. 2017. Model-Based Self-Aware Performance and Resource Management Using the Descartes Modeling Language. *IEEE Transactions on Software Engineering (TSE)* PP, 99 (2017). DOI: http://dx.doi.org/10.1109/TSE.2016.2613863

[4] Samuel Kounev and Alejandro Buchmann. 2006. SimQPN: A Tool and Methodology for Analyzing Queueing Petri Net Models by Means of Simulation. *Perform. Eval.* 63, 4 (2006), 364–394. http://dx.doi.org/10.1016/j.peva.2005.03.004

[5] Samuel Kounev, Nikolaus Huber, Fabian Brosig, and Xiaoyun Zhu. 2016. A Model-Based Approach to Designing Self-Aware IT Systems and Infrastructures. *IEEE Computer* 49, 7 (2016), 53–61.

[6] Philipp Meier, Samuel Kounev, and Heiko Koziolek. 2011. Automated transformation of Palladio component models to queueing Petri nets. In *19th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2011), Singapore*.

[7] Simon Spinner, Giuliano Casale, Xiaoyun Zhu, and Samuel Kounev. 2014. LibReDE: A Library for Resource Demand Estimation. In *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering (ICPE 2014)*. ACM Press, New York, USA, 227–228. http://doi.acm.org/10.1145/2568088.2576093

[8] André van Hoorn, Jan Waller, and Wilhelm Hasselbring. 2012. 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)*. 247–248.

[9] Jürgen Walter, Simon Eismann, Nikolai Reed, and Samuel Kounev. 2017. Architectural Performance Model Extraction as a Service. In *Proceedings of the 2017 Symposium on Software Performance (SSP)*.

[10] Jürgen Walter, Maximilian König, Simon Eismann, and Samuel Kounev. 2016. PAVO: A Framework for the Visualization of Performance Analyses Results. In *Proceedings of the 2016 Symposium on Software Performance (SSP)*.

[11] Jürgen Walter, Dusan Okanovic, and Samuel Kounev. 2017. Mapping of Service Level Objectives to Performance Queries. In *Proceedings of the 2017 Workshop on Challenges in Performance Methods for Software Development (WOSP-C'17)*. ACM.

[12] Jürgen Walter, Christian Stier, Heiko Koziolek, and Samuel Kounev. 2017. An Expandable Extraction Framework for Architectural Performance Models. In *Proceedings of the 3rd International Workshop on Quality-Aware DevOps (QUDOS'17)*. ACM, 6.

[13] Jürgen Walter, Andre van Hoorn, and Samuel Kounev. 2017. Automated and Adaptable Decision Support for Software Performance Engineering. In *Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools*.

[14] Jürgen Walter, Andre van Hoorn, Heiko Koziolek, Dusan Okanovic, and Samuel Kounev. 2016. Asking "What?", Automating the "How?": The Vision of Declarative Performance Engineering. In *7th ACM/SPEC Int. Conf. on Perf. Eng. (ICPE '16)*.