The Slingshot Simulator: An Extensible Event-Driven PCM Simulator

Julijan Katić, Floriment Klinaku and Steffen Becker

Software Quality and Architecture, University of Stuttgart, Germany

Keywords
Palladio Component Model, performance simulator, event-driven architecture, extensibility

1. Context

Architects can use the Palladio Component Model (PCM) to model and evaluate the performance of software systems at design-time [1]. One way of conducting the evaluation is by interpreting the models by so-called performance simulators. Notable PCM simulators are the SimuLizar [2] and EventSim [3].

However, over the years both software performance simulators together with the PCM language itself were enhanced with new features, resulting in large code base that is difficult to extend and maintain. Although SimuLizar, for example, provides many (explicit) extension points for different purposes, the monolithic architecture is resistant against changes, making it complex to introduce new extension points. All in all, we introduce a new simulator for PCM models that bases on the Event-Driven Architecture (EDA) [4] called Slingshot.

2. Objective

The Slingshot simulator is also a PCM simulator and is created with extensibility in mind. We found that an appropriate architecture - in this case the Event-Driven Architecture - can allow us to easily add and remove features into the system . The main objective is to know whether EDA improves extensibility for PCM simulators. For understanding this we compare the current prototypical implementation of Slingshot with the state-of-the-art PCM simulators with respect to extensibility. Each extension should furthermore be able to contribute their own events, making them inherently extensible as well. This work introduces the architecture of the Slingshot simulator that follows the EDA paradigm, the results on extensibility and the current state of development.

SSP 2021: 12th Symposium on Software Performance 2021

st154933@stud.uni-stuttgart.de (J. Katić); floriment.klinaku@iste.uni-stuttgart.de (F. Klinaku); steffen.becker@iste.uni-stuttgart.de (S. Becker)

3. Method

First, we developed the core module of Slingshot after several architecture reviews. This module contains one extension point in which new features can be plugged-in. Furthermore, core meta-events are defined to describe the run-time state of the system, on which extensions can react and behave accordingly. This allows for implementing specific simulation behaviors directly afterwards, such as the standard interpreters (usage model, system and resource interpretation). Finally, the implementation is compared to the other simulators by using specific software metrics. Typically, the software metrics for analyzing extensibility are *cohesion* (such as Lack-of-Cohesion-in-Methods (LCOM)) and *coupling* (such as Coupling-Between-Objects (CBO)), since highly cohesive, loosely coupled and modular systems tend to be extensible. Furthermore, changeability also factors into extensibility. However, these software metrics are defined on a class-level. Because modularity is more important for extensibility, we needed metrics such as EXTCBO (external coupling between object classes) and CC (component cohesion) [5].

For changeability, we counted the number of lines that needed to be changed in the existing code base when adding a certain interpretation behavior in the simulators. For SimuLizar and EventSim, we hypothetically removed the lines responsible for that behavior and looked at what changes were needed.

4. Results

The results show that EDA does improve extensibility without creating further explicit extension points. On one hand, Slingshot requires less configuration when adding new extensions. Furthermore, as long as the modules remain highly cohesive, Slingshot has looser coupling and is highly changeable. On the other hand, even though SimuLizar has rather low EXTCBO, it is hardly adaptable to new features since it provides only a finite set of extension points. Its changeability also suffered by the scenario. The results, therefore, suggest that extensibility is affected if the right software architecture was chosen, and that it can be improved by using EDA.

It is important to note that the current results only show the aspect of adding another interpreter into the system. However, extensibility is subjective to the developers and their skill-set, as well as documentation [6]. In the future, a case study should be conducted to improve the analysis. Thus, the results only suggest that Slingshot is currently *more* extensible.

At the moment, Slingshot is still under development. As of now, only the standard interpretation is featured. Later, it will also be able to monitor and simulate self-adaptive systems.

5. Conclusion

We create a new PCM simulator named Slingshot to achieve high extensibility. The results suggest that Slingshot is extensible regarding adding of new simulation capabilities by simply listening to existing events and contributing new events. However, further analysis is needed to check whether extensibility is achieved in all aspects, and whether Slingshot suffers in areas where SimuLizar and EventSim are providing extension points. Also, other metrics such as performance and validity are still being investigated.

References

[1] R. Reussner, S. Becker, E. Burger, J. Happe, M. Hauck, A. Koziolek, H. Koziolek, K. Krogmann, M. Kuperberg, The Palladio Component Model, Technical Report 14, Karlsruher Institut für Technologie (KIT), 2011. doi:10.5445/IR/1000022503.

[2] M. Becker, S. Becker, J. Meyer, Simulizar: Desing-time modeling and performance analysis of self-adaptive systems, in: S. Kowalewski, B. Rumpe (Eds.), Software Engineering 2013, Gesellschaft für Informatik e.V., Bonn, 2013, pp. 71–84.

[3] P. Merkle, J. Henß, Eventsim - an event-driven palladio software architecture simulator, in: Palladio Days 2011. Proceedings, 17-18 November 2011, FZI Forschungszentrum Informatik, Karlsruhe, Germany. Ed.: S. Becker, number 32 in Karlsruhe Reports in Informatics (früher: Interner Bericht. Fakultät für Informatik, Karlsruher Institut für Technologie), 2011, pp. 15–22. doi:10.5445/IR/1000025188.

[4] R. Bruns, J. Dunkel, Event-Driven Architecture, Springer, 2010. doi:10.1007/978-3-642-02439-9.

[5] T. Vernezza, G. Granatella, G. Succi, L. Benedicenti, M. Mintchev, Defining metrics for software components, Proceedings of. World Multi-conference on Systematics, Cubernetics and Informatics 11 (2000) 16–23. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.205.8052&rep=rep1&type=pdf.

[6] A. Kelly, The philosophy of extensible software, The Internet's Coming Silent Spring (2002).