# Supporting and Verifying Transient Behavior Specifications in Chaos Engineering [Extended Abstract]

Denis Zahariev
University of Stuttgart, Stuttgart

Sebastian Frank
sebastian.frank@uni-hamburg.de
University of Hamburg, Hamburg

Alireza Hakamian
mir-alireza.hakamian@iste.uni-stuttgart.de
University of Stuttgart, Stuttgart

André van Hoorn
andre.van.hoorn@uni-hamburg.de
University of Hamburg, Hamburg

## 1 Context and Problem

Chaos Engineering [2] is an approach for assessing the resilience of software systems, i.e., their ability to withstand unexpected events, adapt accordingly, and continue providing functionality. An integral part of the approach is continuous experimentation, expressed in continuously executing so-called Chaos Experiments. When applied, the traditional Chaos Engineering approach only verifies whether the system is in a steady state without providing information about the time between the state changes, e.g., the recovery of the system. The experimentation process conceptually does not allow the specification of hypotheses regarding the transient behavior, i.e., the behavior experienced during the transition between steady states after a failure has been injected.

## 2 Objective

Our goal is to study how the Chaos Engineering process can include verification of transient behavior requirements. We aim to extend the Chaos Engineering approach and tooling to support the specification of transient behavior hypotheses and their verification. Moreover, we also aim to provide assistance to the actual users of our extended Chaos Engineering approach and make the tooling compatible with our existing approaches for resilience assessment [7]. Thus, we evaluate our approach regarding the correctness in and outside of Chaos Experiments.

## 3 Method

We extend the Chaos Experiment process with a transient behavior specification using formalisms such as Metric Temporal Logic (MTL) and Property Specification Patterns (PSP) [1]. Consequently, we study the inclusion of runtime verification into the Chaos Experimentation process. Based on a comparison of state-of-the-art Chaos Engineering tools, we select Chaos Toolkit [6] for a prototypical extension. Furthermore, we conducted interviews with three Chaos Engineering experts and practitioners. To create the extended Chaos Engineering approach, we combined the requirements elicited during the interviews into a concept, which is then implemented into a prototype.

We conduct the correctness evaluation with data provided by a benchmark generator for MTL monitoring tools [3] using the past and future MTL formulas of PSPs. We execute chaos experiments on a simplified testing system representing an actual industry system and use our approach to verify transient behavior specifications. Finally, we run simulations of chaos experiments using our simulator MiSim [5], which is designed for resilience assessment, and use our approach to verify transient behavior specifications based on MiSim's outputs.

## 4 Result

The primary outcome of this work is an approach and a browser-based tool that provides assistance in specifying transient behavior, creating chaos experiment specifications for Chaos Toolkit, and verifying the transient behavior against retrieved monitoring data. Furthermore, it has basic capabilities for the visualization of results.

The tool can correctly verify all the 160 specifications in the benchmark, covering four different PSPs with four different scopes, positive and negative outcomes, as well as future and past MTL formulas. Furthermore, the tool correctly verified our created transient behavior specifications in data collected from three chaos experiments and one simulation run.

## 5 Talk Outline and Additional Resources

In our talk, we will present our approach and tool for verifying transient behavior specifications in Chaos Engineering. In addition, we will elaborate on selected results from our experiments. Since the project is embedded into a research endeavor already presented at SSP21 [4], we will also outline the role of this work

in the ongoing endeavor. Thus, we will show preliminary concepts and outcomes of follow-up works, e.g., regarding the optimization of verified requirements.

## 6 Acknowledgments

The authors thank the German Federal Ministry of Education and Research (dqualizer project and Software Campus 2.0 — Microproject: DiSpel, FKZ: 01IS17051) for supporting this work.

## References

[1] M. Autili et al. "Aligning qualitative, real-time, and probabilistic property specification patterns using a structured english grammar". In: *IEEE Transactions on Software Engineering* 41.7 (2015), pp. 620–638.

[2] A. Basiri et al. "Chaos Engineering". In: *IEEE Software* 33 (Jan. 2016), pp. 1–1.

[3] D. Ulus. "Timescales: A benchmark generator for MTL monitoring tools". In: *International Conference on Runtime Verification*. Springer. 2019, pp. 402–412.

[4] S. Frank et al. "Scenario-Based Elicitation, Specification, and Comprehension of Transient Software Behavior". In: *SSP 2021* (2021).

[5] L. Wagner et al. "MiSim—A Lightweight and Extensible Simulator for a Scenario-Based Resilience Evaluation of Microservice Architectures". In: *SSP 2021* (2021).

[6] *Chaos Toolkit*. 2022. URL: `https://github.com/chaostoolkit`.

[7] S. Frank et al. "Interactive Elicitation of Resilience Scenarios Based on Hazard Analysis Techniques". In: *Lecture Notes in Computer Science*. Vol. 13365. (ECSA'21 post-proceedings; in press). Springer, 2022.