

# Quantifying the Potential to Automate the Synchronization of Variants in Clone-and-Own – Summary

Alexander Schultheiß[1], Paul Maximilian Bittner[2], Thomas Thüm[3], Timo Kehrer[4]

**Abstract:** We report about a recent empirical study on variant synchronization in clone-and-own, originally published at the 38th IEEE International Conference on Software Maintenance and Evolution (ICSME) 2022 [Sc22]. In clone-and-own, a new variant of a software system is created by copying and adapting an existing one. While it is flexible, clone-and-own causes high maintenance effort in the long run as cloned variants evolve in parallel; certain changes, such as bug fixes, need to be propagated between variants. A recent line of research proposes to automate such synchronization tasks when migration to a software product line is not feasible. However, it is yet unclear how far this synchronization can actually be pushed. We present an empirical study in which we quantify the potential to automate the synchronization of variants in clone-and-own. We simulate automated variant synchronization using the history of BusyBox, a real-world multi-variant software system. Our results indicate that existing patching techniques propagate changes with an accuracy of up to 85%, if applied consistently from the start of a project. This can be even further improved to 93% by exploiting lightweight domain knowledge about which features are affected by a change, and which variants implement affected features.

**Keywords:** clone-and-own, variant synchronization, version control, software product lines

## Summary

Today's software is often released in multiple variants to meet varying requirements. While there are systematic approaches to managing variability, such as software product lines where all variants are managed using an integrated platform, these approaches are not feasible for all projects. Instead, developers fall back to using clone-and-own, creating a new variant of a software system by copying and adapting an existing one (e. g., using branching/forking capabilities of a version control system). This way, new variants are created ad-hoc and without requiring upfront investments or knowledge about future variants. In the long term, however, clone-and-own projects suffer from ever-increasing maintenance costs. For example, if a bug is discovered and fixed in one variant, it is often unclear which other variants are affected by the same bug and how it should be fixed in these variants.

Researchers started to explore the continuum between ad-hoc clone-and-own and software product lines to reduce the burden on developers. In our project *VariantSync* [Ke21], we

[1] Humboldt-Universität zu Berlin, Germany, alexander.schultheiss@informatik.hu-berlin.de
[2] University of Ulm, Germany, paul.bittner@uni-ulm.de
[3] University of Ulm, Germany, thomas.thuem@uni-ulm.de
[4] University of Bern, Switzerland, timo.kehrer@inf.unibe.ch

propose to manage the development and maintenance of cloned variants by propagating changes of interest between them, thereby keeping cloned variants synchronized. To this end, we advocate to collect additional (lightweight) domain knowledge on cloned variants and software changes, which might be leveraged by change propagation techniques. However, it is yet unclear how far this synchronization can actually be pushed.

In this work [Sc22], we quantify the potential to automate the synchronization of variants in clone-and-own through an empirical study. Our study covers three aspects of patch-based variant synchronization. First, to gain insight on the difficulties of automated change propagation, we investigate how often propagating a change via patching succeeds or fails, depending on different levels of patch granularity (i. e., commit-, file-, and line-level patches). Second, we examine the correctness of automated patching by analyzing the outcome of each synchronization scenario. Third, we investigate the potential to improve the correctness of automated synchronization when developers document lightweight domain knowledge. Specifically, we employ lightweight domain knowledge about which features are affected by a change and which variants implement affected features – knowledge that is generally assumed to be available but typically undocumented in clone-and-own.

We inspect almost half a billion patch scenarios derived from a large-scale real-world system (BusyBox). We find that the majority of patches is applicable automatically, even when propagating changes blindly across variants. Blind patching produces correct results in the majority of cases with an accuracy of 85% and precision of 92%. This shows that the very applicability of patches is a useful indicator for determining if a target variant should actually receive a patch or not. Furthermore, we confirm the hypothesis that gathering lightweight domain knowledge might prove useful for automated synchronization. If automated patching with lightweight domain knowledge is applied from the start of a project, variants can be synchronized with high accuracy (93%) and almost perfect precision (97%).

## Data Availability

The publication will be accessible under the DOI 10.1109/ICSME55016.2022.00032. A preprint can be found on our website (`https://seg.inf.unibe.ch/papers/SBTK_ICSME22.pdf`). Our artifact is available on Github (`https://github.com/VariantSync/SyncStudy`) and Zenodo (DOI: 10.5281/zenodo.7025599).

## Bibliography

[Ke21] Kehrer, Timo; Thüm, Thomas; Schultheiß, Alexander; Bittner, Paul Maximilian: Bridging the Gap Between Clone-and-Own and Software Product Lines. In: Proc. Int'l Conf. on Software Engineering (ICSE). IEEE, pp. 21–25, 2021.

[Sc22] Schultheiß, Alexander; Bittner, Paul Maximilian; Thüm, Thomas; Kehrer, Timo: Quantifying the Potential to Automate the Synchronization of Variants in Clone-and-Own. In: Proc. Int'l Conf. on Software Maintenance and Evolution (ICSME). IEEE, 2022. To appear.