



## **Intention-Based Integration of Software Variants**

Downloaded from: <https://research.chalmers.se>, 2026-05-17 18:20 UTC

Citation for the original published paper (version of record):

Lillack, M., Stănciulescu, Ș., Hedman, W. et al (2020). Intention-Based Integration of Software Variants. Lecture Notes in Informatics, P-300: 57-58. [http://dx.doi.org/10.18420/SE2020\\_15](http://dx.doi.org/10.18420/SE2020_15)

N.B. When citing this work, cite the original published paper.

## Intention-Based Integration of Software Variants

Max Lillack,<sup>1</sup> Stefan Stanculescu,<sup>2</sup> Wilhelm Hedman,<sup>3</sup> Thorsten Berger,<sup>1</sup> Andrzej Wasowski<sup>4</sup>

**Abstract:** This abstract summarizes our paper with the same title published at the 41st International Conference on Software Engineering (ICSE) 2019 [Li19].

**Keywords:** software variants; software re-engineering; software integration; software product lines

Cloning is a simple way to create new variants of a system. While cheap at first, it increases maintenance cost in the long term. Eventually, the cloned variants need to be integrated into a configurable platform. Such an integration is challenging: it involves merging the usual code improvements between the variants, and also integrating the variable code (features) into the platform. As such, variant integration differs from traditional software merging, which does not produce or organize configurable code, but creates a single system that cannot be configured into variants. In practice, variant integration requires fine-grained code edits, performed in an exploratory manner, in multiple iterations. Unfortunately, little tool support exists for integrating cloned variants.

In this work, we show that fine-grained code edits needed for integration can be alleviated by a small set of integration *intentions*—domain-specific actions declared over code snippets controlling the integration. Developers can interactively explore the integration space by declaring (or revoking) intentions on code elements. We contribute the intentions (e.g., ‘keep functionality’ or ‘keep as a configurable feature’) and the IDE tool INCLINE, which implements the intentions and five editable views that visualize the integration process and allow declaring intentions producing a configurable integrated platform. In a series of experiments, we evaluated the completeness of the proposed intentions, the correctness and performance of INCLINE, and the benefits of using intentions for variant integration. The experiments show that INCLINE can handle complex integration tasks, that views help to navigate the code, and that it consistently reduces mistakes made by developers during variant integration.

---

<sup>1</sup> Leipzig University, Germany

<sup>2</sup> ABB Corporate Research Center, Switzerland

<sup>3</sup> Chalmers | University of Gothenburg, Sweden

<sup>4</sup> IT University of Copenhagen, Denmark

## References

- [Li19] Lillack, Max; Stanculescu, Stefan; Hedman, Wilhelm; Berger, Thorsten; Wasowski, Andrzej: Intention-Based Integration of Software Variants. In: 41st International Conference on Software Engineering (ICSE). 2019.