Do Developers Benefit from Recommendations when Repairing Inconsistent Design Models? a Controlled Experiment

Luciano Marchezan[1], Wesley Assunção[1], Gabriela Michelon[1], and Alexander Egyed[1]

[1]Johannes Kepler University Linz, Institute of Software Systems Engineering, Altenberger Straße 69, 4040 Linz, Austria

Keywords: Controlled Experiment; Consistency Checking; Repair Recommendations

Repairing inconsistent models is an error-prone and laborious task that requires a considerable amount of time and effort from developers [Jo22; Ma23a; Tr22]. Hence, repair recommendation (RR) approaches can be applied to maintain consistency and reduce the time and effort required for the repair task, mainly on UML models [Ba16]. Approaches proposing the use of RRs have been evaluated in a variety of scenarios in terms of scalability, correctness, and minimalism [Ma22a; Ma22b]. However, most studies in the RR field do not focus their evaluations on the perspective of developers regarding the provision of RRs. These evaluations are important to analyze the implications of applying RRs strategies from the perspective of its end-users, i.e., developers. In addition, as developers have different experiences and preferences, ranking or automatically executing RRs to fix inconsistencies may change the models in a way not desired by the developer. Hence, it is important to understand the developers' preferences when repairing models before applying these approaches in industrial settings.

We address the aforementioned limitations by conducting an experiment with developers, driven by three research questions (RQ): **RQ1**. Do developers benefit from recommendations when repairing inconsistent design models? **RQ2**. How do developers perceive the use of repair recommendations when repairing inconsistent design models? **RQ3**. Do developers have preferred recommendations when repairing inconsistent design models? The data used to answer the RQs was collected by analyzing developers repairing inconsistent models with and without the provision of RRs. The sample was composed of 24 M.Sc./Ph.D. students with varied software development experience. To answer RQ1, we measure how the use of RRs can bring benefits to developers in terms of effectiveness (i.e., inconsistencies fixed) and efficiency (i.e., the time required) when repairing design models compared to when RRs are not provided. For RQ2, we asked developers to give us feedback regarding the difficulty and their confidence when repairing the models. For RQ3, we ask the developers to select and rank RRs to understand their preferences for different tasks.

doi:10.18420/sw2024_29

Results show that the provision of RRs benefits developers by improving their effectiveness by 37.63% (p-value of 0.04) and efficiency by 29.81% (p-value of 0.17) in comparison to not having RRs (RQ1). In more simple tasks, however, RRs reduce the efficiency of developers. These findings evidence that RRs approaches can bring benefits to developers in more complex tasks. Furthermore, the perceived difficulty and confidence of developers when RRs are given are similar without RRs (RQ2). We also observed that developers do not have an ideal RR for a given inconsistency, but rather have different preferences regarding how to repair a model (RQ3). Moreover, in some contexts, any RR may be considered not applicable. These findings highlight the importance of having developers' feedback when repairing models since they have different preferences. Such results also indicate that applying automatic RRs in models by selecting the "most suited" RR may not be ideal, as developers do not have the same opinion about what is the "most suited" RR.

The original paper is available at [Ma23b] with an online appendix at `https://sites.google.com/view/rrexperiment`.

References

[Ba16] Bashir, R. S.; Lee, S. P.; Khan, S. U. R.; Chang, V.; Farid, S.: UML models consistency management: Guidelines for software quality manager. IJIM 36/6, Part A, pp. 883–899, 2016, ISSN: 0268-4012.

[Jo22] Jongeling, R.; Ciccozzi, F.; Carlson, J.; Cicchetti, A.: Consistency Management in Industrial Continuous Model-Based Development Settings: A Reality Check. SoSym 21/4, pp. 1511–1530, Aug. 2022, ISSN: 1619-1366.

[Ma22a] Marchezan, L.; Assuncao, W. K. G.; Kretschmer, R.; Egyed, A.: Change-Oriented Repair Propagation. In: ICSSP. ACM, pp. 82–92, 2022, ISBN: 9781450396745.

[Ma22b] Marchezan, L.; Kretschmer, R.; Assunção, W. K.; Reder, A.; Egyed, A.: Generating repairs for inconsistent models. SoSym/, pp. 1–33, 2022.

[Ma23a] Marchezan, L.; Assunção, W. K. G.; Herac, E.; Keplinger, F.; Egyed, A.; Lauwerys, C.: Fulfilling Industrial Needs for Consistency Among Engineering Artifacts. In: 45th International Conference on Software Engineering (ICSE) - Software Engineering in Practice. Pp. 1–12, 2023.

[Ma23b] Marchezan, L.; Assunção, W. K. G.; Michelon, G. K.; Egyed, A.: Do Developers Benefit from Recommendations when Repairing Inconsistent Design Models? a Controlled Experiment. In: 27th International Conference on Evaluation and Assessment in Software Engineering (EASE). Pp. 1–10, 2023.

[Tr22] Tröls, M. A.; Marchezan, L.; Mashkoor, A.; Egyed, A.: Instant and global consistency checking during collaborative engineering. Software and Systems Modeling/, pp. 1–27, 2022.