Choosing Abstraction Levels for Model-Based Software Debugging: A Theoretical and Empirical Analysis for Spreadsheet Programs (Abstract Reprint)

Patrick Rodler[1], Birgit Hofer[2], Dietmar Jannach[1], Iulia Nica[1], Franz Wotawa[2]

[1]University of Klagenfurt, Austria
[2]Graz University of Technology, Austria

Abstract Reprint. This is an abstract reprint of the journal article by Rodler, Hofer, Jannach, Nica, and Wotawa (2025).

Abstract

Model-based diagnosis is a generally applicable, principled approach to the systematic debugging of a wide range of system types such as circuits, knowledge bases, physical devices, or software. Based on a formal description of the system, it enables precise and deterministic reasoning about potential faults responsible for observed misbehavior. In software, such a formal system description can often even be extracted from the buggy program fully automatically. As logical reasoning is central to diagnosis, the performance of model-based debuggers is largely influenced by reasoning efficiency, which in turn depends on the complexity and expressivity of the system description. Since highly detailed models capturing exact semantics often exceed the capabilities of current reasoning tools, researchers have proposed more abstract representations.

In this work, we thoroughly analyze system modeling techniques with a focus on fault localization in spreadsheetsone of the most widely used end-user programming paradigms. Specifically, we present three constraint model types characterizing spreadsheets at different abstraction levels, show how to extract them automatically from faulty spreadsheets, and provide theoretical and empirical investigations of the impact of abstraction on both diagnostic output and computational performance. Our main conclusions are that (i) for the model types, there is a trade-off between the conciseness of generated fault candidates and computation time, (ii) the exact model is often impractical, and (iii) a new model based on qualitative reasoning yields the same solutions as the exact one in up to more than half the cases while being orders of magnitude faster.

Due to their ability to restrict the solution space in a sound way, the explored model-based techniques, rather than being used as standalone approaches, are expected to realize their full potential in combination with iterative sequential diagnosis or indeterministic but more performant statistical debugging methods.

References

Rodler, P.; Hofer, B.; Jannach, D.; Nica, I.; and Wotawa, F. 2025. Choosing abstraction levels for model-based software debugging: A theoretical and empirical analysis for spreadsheet programs. *Artificial Intelligence*, 348: 104399.