
Type of paper: Radical Method

Correlation to the real-world setting: References trends in programming like increasing abstraction, and also widely acknowledged problems like the cost of developing software and the lack of quality in software implementations.

What is the hypothesis: Development tools should come with a library of transformations that have been proven to be reliable. Developers should code by applying these transformations to optimize an abstract specification. This ensures that the code matches the specification while meeting the performance goals for the software.

Validation: The authors provide a toy example in a high-level specification language. I don’t see a definition of the language. They then refine this specification to eliminate loops and write code that looks similar to C.

Result: A chain from high-level code to low-level code. I don’t see proofs of the transformations.

Do you believe the result, and why? I believe that the missing bits of the proof could be completed in this specific case. However, I’m skeptical that developers and users can agree on an unambiguous high-level specification for most problems.