On “Methodological Irregularities in Programming Language Research”

Jeremy Gibbons

14th October 2017

Stefik and Hanenberg recently published an opinion piece on “methodological irregularities in programming-language research” [1] in *IEEE Computer*. They criticize programming language (PL) design conferences such as PLDI, OOPSLA, ICFP, and ECOOP for a lack of “rigorous evidence standards like those in other sciences”—standards such as randomized controlled trials for evaluating the effectiveness of an intervention. They conclude by calling for the imposition of strict reporting standards in software engineering conference and journals, like the *Consolidated Standards of Reporting Trials* (CONSORT) statement [2] in medicine and the *What Works Clearinghouse* (WWC) guidelines [3] in education.

Stefik and Hanenberg’s recommendations are reasonable, for a certain class of software engineering research—specifically, for research aiming to show that a given intervention, such as the adoption of a particular language or technique, has certain economic or pedagogical benefits. Randomized controlled trials are the gold standard of evidence for this kind of claim, and repeatability requires careful attention to sample size and selection, control of confounding factors, and so on. Robust empirical claims require robust empirical evidence.

However, it is narrow-minded to expect all PL research to follow this empirical pattern. Not all scientific claims concern the effectiveness of an intervention on human subjects, and so not all papers are suitable targets for standards such as CONSORT and WWC. In particular, it is perverse to criticize conferences such as ICFP for a “lack [of] empirical foundation”, when the papers published there mostly do not make empirical claims. One might as well criticize Einstein [4], Turing [5], and Watson and Crick [6] for “methodological irregularities”.

There is much more to PL research than empirical claims of the kind that Stefik and Hanenberg have in mind—for example, mathematical semantics of language features, verification and proof of correctness, type systems to allow the formal statement and certain properties of programs, static and dynamic analysis to check those properties, systems building and engineering design, compiler and performance optimizations, implementation techniques, and so on. For a longer discussion, see for example the *PL Enthusiast* blog post [7].

Published (after editing) in the “Letters” column in IEEE Computer 51(4):4–7, April 2018, doi `10.1109/MC.2018.2141027`.

References

[1] Andreas Stefik and Stefan Hanenberg. Methodological irregularities in programming-language research. *IEEE Computer*, 50(8):60–63, 2017.

[2] The CONSORT statement. `http://www.consort-statement.org/`.

[3] Institute of Education Sciences, US Department of Education. What Works Clearinghouse: Procedures and standards handbook. `http://ies.ed.gov/ncee/wwc/Docs/referenceresources/wwc_procedures_v3_0_standards_handbook.pdf`, 2014.

[4] Albert Einstein. Ist die Trägheit eines Körpers von seinem Energieinhalt abhängig? (Does the inertia of a body depend upon its energy content?). *Annalen der Physik*, 18(13):639–641, 1905.

[5] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, Series 2*, 42:230–265, 1937.

[6] James D. Watson and Francis H. Crick. A structure for deoxyribose nucleic acid. *Nature*, (4356):737–738, April 1953.

[7] Michael Hicks. What is PL research and how is it useful? `http://www.pl-enthusiast.net/2015/05/27/what-is-pl-research-and-how-is-it-useful/`, May 2015.