Adventures in NICAD:
A Ten-Year Retrospective

MIP Abstract

Chanchal K. Roy
University of Saskatchewan
Saskatoon, SK, Canada
canchal.roy@usask.ca

James R. Cordy
Queen’s University
Kingston, ON, Canada
cordy@cs.queensu.ca

ABSTRACT

Based on the simple, naive idea of text-line differencing of pretty-printed code, at ICPC 2008 we introduced NICAD [5], the first code clone detector explicitly aimed at finding intentional “near-miss” (Type 3) clones. Using the TXL [2] parser to identify and pretty-print all instances of a code unit of interest (functions, blocks, etc.), NICAD provides several ways to pre-process the code before comparison, including flexible formatting, renaming, normalization and abstraction, making it suitable for finding all kinds of clones in a wide range of different applications. In this talk we will outline the journey from that initial naive idea to an efficient, scalable, flexible clone detection tool that handles more than ten different languages with high accuracy in both precision and recall [8]. Along the way we will highlight our experience in tuning our initial prototype to production speed and scalability [4], we will review its application in a range of large-scale clone experiments [3, 6, 7], and describe its evolution to handle new domains such as subsystem clones in graphical models [1]. Finally, we will close with new methods based on NICAD [9, 10] and its lessons for clone detection research in the future.

ACKNOWLEDGMENTS

The authors wish to acknowledge NSERC, the Natural Sciences and Engineering Research Council of Canada, for their continuing support over the many years of the NICAD project.

REFERENCES