

# Adventures in NICAD: A Ten-Year Retrospective

MIP Abstract

Chanchal K. Roy  
University of Saskatchewan  
Saskatoon, SK, Canada  
chanchal.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.

## BIOGRAPHIES

Chanchal K. Roy is Associate Professor of Software Engineering at the University of Saskatchewan, Canada. He is the co-lead of the Big Data Analytics group of an NSERC Canada First Research Excellence Fund (CFREF) on Food security. As the co-author of the widely used NICAD code clone detection system, he has published more than 140 refereed publications that have been cited more than 3600 times. His contributions to the software maintenance community, and particularly to the software clones community, have been highly influential, recently winning Most Influential Paper awards at both SANER 2018 and ICPC 2018. Dr. Roy was a vision keynote speaker at WCRE/CSMR 2014 on software clones, and a keynote speaker at IWSC 2018 and IEEE R10HTC 2018. He serves widely on the program committees of major software engineering conferences such as ICSE, ICSME, SANER, MSR, ICPC and SCAM, and has chaired several.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).  
ICPC '18, May 27–28, 2018, Gothenburg, Sweden  
© 2018 Copyright held by the owner/author(s).  
ACM ISBN 978-1-4503-5714-2/18/05...\$15.00  
<https://doi.org/10.1145/3196321.3196325>

James R. Cordy is Professor and past Director of the School of Computing at Queen's University in Kingston, Ontario, Canada, and recent Director of the NSERC CREATE Graduate Specialization in Ultra-Large Scale Software Systems. From 1995 to 2000 he was Vice President and Chief Research Scientist at Legasys Corporation, a software technology company specializing in legacy software system analysis and renovation. As leader of the TXL source transformation project with hundreds of academic and industrial users worldwide, he has been involved in software analysis and transformation systems for more than 30 years. He has published more than 200 refereed contributions in software engineering, programming languages and artificial intelligence, three of which have recently won ten-year Most Influential Paper awards. The 2016 winner of the Queen's University Prize for Excellence in Research, he is an ACM Distinguished Scientist, a Senior Member of the IEEE, and an IBM CAS Faculty Fellow.

## 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

- [1] M. H. Alalfi, J. R. Cordy, T. R. Dean, M. Stephan, and A. Stevenson. 2012. Models are code too: Near-miss clone detection for Simulink models. In *28th IEEE International Conference on Software Maintenance, ICSM 2012*. 295–304.
- [2] J.R. Cordy. 2006. The TXL source transformation language. *Science of Computer Programming* 61, 3 (2006), 190–210.
- [3] J. R. Cordy. 2011. Exploring large-scale system similarity using incremental clone detection and live scatterplots. In *19th IEEE International Conference on Program Comprehension, ICPC 2011*. 151–160.
- [4] J. R. Cordy and C. K. Roy. 2014. Tuning research tools for scalability and performance: the NICAD experience. *Science of Computer Programming* 79, 1 (2014), 158–171.
- [5] C.K. Roy and J.R. Cordy. 2008. NICAD: Accurate detection of near-miss intentional clones using flexible pretty-printing and code normalization. In *16th IEEE International Conference on Program Comprehension, ICPC 2008*. 172–181.
- [6] C. K. Roy and J. R. Cordy. 2010. Are scripting languages really different?. In *4th International Workshop on Software Clones, IWSC 2010*. 17–24.
- [7] C. K. Roy and J. R. Cordy. 2010. Near-miss function clones in open source software: an empirical study. *Journal of Software Maintenance and Evolution: Research and Practice* 22, 3 (2010), 165–189.
- [8] J. Svajlenko and C. K. Roy. 2015. Evaluating clone detection tools with Big-CloneBench. In *31st IEEE International Conference on Software Maintenance and Evolution, ICSME 2015*. 131–140.
- [9] Jeffrey Svajlenko and Chanchal K. Roy. 2017. Fast and flexible large-scale clone detection with CloneWorks. In *39th International Conference on Software Engineering Companion, ICSE-C 2017*. 27–30.
- [10] M. S. Uddin, C. K. Roy, and K. A. Schneider. 2013. SimCad: An extensible and faster clone detection tool for large scale software systems. In *21st IEEE International Conference on Program Comprehension, ICPC 2013*. 236–238.