Clone wars—finding buggy code copies

June 11 2018, by Kris Foster

Chanchal Roy, associate professor in the Department of Computer Science. Credit: Kris Foster

Code is ubiquitous and most industries around the world rely on code-based software to keep day-to-day operations running, said Chanchal Roy, associate professor in the Department of Computer Science.

"The simplest functions use code, and bad code can have a massive impact," said Roy, who joined the College of Arts and Science in 2009. "Unfortunately, the way developers copy code can result in lots of bugs or errors, something my research addresses."

It is common practice for software developers to copy, paste and modify a fragment of existing code to suit the task or tool they are working on. This is called cloning, and the resulting code from the copy-and-paste process is, of course, called a clone.

"There are valid reasons why cloning is so common," said Roy, whose research is supported by a Natural Sciences and Engineering Research Council of Canada Accelerator Grant. "It saves time, there is low risk in using stable code, and it results in faster development. There is no need to reinvent the wheel."

The problem, Roy is quick to point out, is that often cloning code results in cloning unknown "bugs" as well, and these errors can spread quickly.

"If you have a bug in the original code, you are copying errors over and over again," he said. "Even if you find one instance of the bug, it is nearly impossible to find all of them … which results in a lot of industries using outdated code over new code that potentially has bugs."

In part because of the issues related to cloning and the resulting buggy clones, up to 85 per cent of the cost of software development can go towards software maintenance, including clone detection.

"It is a double-edged sword," said Roy. "Cloning is common because of the benefits to programmers, but clones can carry bugs that are also really troublesome."

Clone detection, an area in which Roy has dedicated a lot of research

time, means finding similar code fragments in order to resolve bug issues. In its simplest form, it is like doing a document search for specific words. In its most complex form, it is like searching for a needle in a haystack, especially if the original code has been modified (which is the most common form of cloning) and is in a program containing millions of lines of code.

To address this issue, Roy and his research collaborator James Cordy of Queen's University have developed a number of clone detection systems that search for similar fragments of code. There are two main criteria needed for a good clone detection system: precision, which is the ability to detect clones correctly; and recall, a term referring to the percentage of clones detected out of the total number of clones present. Roy and Cordy have developed the first clone detection system, called NICAD, that excels in both precision and recall.

"Once we define what similarities to search for, NICAD can detect modified clones," Roy said, noting that a great amount of human testing, including vetting over nine million cloned fragments, has gone towards ensuring the clone detection system is accurate.

Through his evaluation of clone detection, Roy has also become a world leader in the area of benchmarking clone detection tools with the development of the BigCloneBench tool.

The potential of Roy's clone detection systems and benchmarking work is not going unnoticed. Roy and Cordy have recently received two Most Influential Paper awards, in recognition of the "lasting impact of contributions made within the previous 10 years." Their work on benchmarking and NICAD were recognized by the International Conference on Software Analysis, Evolution and Reengineering, and the International Conference on Program Comprehension, respectively.

Looking ahead to the next decade, Roy said he would like to develop a "safe cloning system" that not only detects corrupt clones, but is also able to advise on how to fix bugs in the system, or even remove them automatically.

"This has the potential to save a lot of time and money, but I am not sure I can do this even in the next 20 years," said Roy with a slight smile and laugh.

Provided by University of Saskatchewan