



Illustration by Fred Reibin

Building on a Legacy

Working with users to revitalize the CRHM hydrological model

Dr. Banani Roy, Director, Interactive Software Engineering and Analytics Lab, University of Saskatchewan

One wintery day in early 2018, I received an email from Computer Science Professor Kevin Schneider. He had been contacted by Global Water Futures Director John Pomeroy with an unusual request: to find a way to rescue a much-valued modelling tool so John's team could continue its studies of the mountain and arctic water sources that feed the rivers of the Canadian West. Hydrology was a new area for me, but I could see that this was an urgent task.

CRHM is a cold region hydrological modelling scientific software that was developed in 1998 and has since been used worldwide by government agencies, universities, and corporations. Initially devised to integrate numerical algorithms derived from hydrological process observations, the software predicts values of environmental variables based on historical data, relieving the need for expensive and time-consuming fieldwork. Just think, a solution where environmental scientists don't need to go to the top of a mountain to collect samples in harsh winter conditions.

The issue was that by 2018, the software developed using the Borland C++ compiler- was becoming outdated and there was no way to provide ongoing support for users and developers. Migration to a modern software development environment had become essential. When I got the source code of CRHM, I thought, "Well, we have inherited 73,000 lines of spaghetti code: how am I going to make it bread and butter?" Moreover, since neither my team nor myself had much background in hydrology, the code often looked mysterious. It was a daunting task for one team; but, as the saying goes, for every problem there is a most simple solution.

CRHM was originally written over the course of three months by a single engineer, Tom Brown. Impressive as this might be, CRHM did not come with any architectural documentation to describe how the system was designed.

To improve the system over the years, Tom responded to the needs of CRHM users as they arose, but at different points in time and with no guiding commentary. To add to the mystery, no version control system was used to keep track of the source code changes. In many cases the code looked to us like spaghetti, with tangled and jumbled concepts. For example, the business logic and user interface code were intertwined. There were many versions of CRHM and Tom was not sure which version was suitable for migration.

The solution to straightening out the spaghetti code was to not do it alone. Our development team met with Tom, John, and other end users to understand the 'hydro-logic' behind CRHM. We learned a lot through these meetings about which features were essential to carry forward into the modern version. In these meetings I took opportunities to explain to them the different technological options that we could adopt to migrate CRHM. Together, we worked on bridging the gap between the computer scientists and hydrology modelers.

I think one of the easiest but most useful tasks in our migration process was putting CRHM on GitHub, providing a version control system that allows different stakeholders to view and contribute to the repository. So far, we have 10 new contributors to the migration effort, some of whom are junior researchers including undergraduate interns, research assistants, graduate students, and postdocs with fresh perspectives on the process. In April 2020 we launched the new CRHM: 1.0. Making the model open source for collaborative software development made it possible to release a more stable version, CRHM 1.3 in August 2022.

Despite this benchmark, software development is an ongoing process. I look forward to what comes next as my team continues to collaborate with hydrology modelers.