Memorandum

To: Brian Macpherson (Colorado Water Conservation Board),
Carolyn Kemp (CWCB),
Mary Halstead (Colorado Division of Water Resources)
From: Steve Malers (Open Water Foundation, OWF)
Subject: Notes on RiverWare
Date: February 3, 2019

Contents

Introduction

This provides notes about RiverWare, in order to provide insight on programming language, development environment, and development team, in order to inform the evaluation of whether StateMod should be converted to a different language. These notes are based on a conversation between Steve Malers of the Open Water Foundation and Edith Zagona and Bill Oakley of the Center for Advanced Decision Support for Water and Environmental Systems (CADSWES) on January 18, 2019.

Background

RiverWare (see https://www.colorado.edu/cadswes/creative-works/riverware) is an advanced river basin simulation tool that is developed by CADSWES (see https://www.colorado.edu/cadswes/). RiverWare has received substantial funding from major federal government agencies, in particular the US Bureau of Reclamation, and has been applied to develop simulation models for basins throughout the US. RiverWare in particular has advanced reservoir simulation features.

Technical Notes

The following are notes on specific technical topics.

Programming Language

RiverWare is coded primarily in Microsoft Visual Studio C++ and Commercial Qt 5.5.1 for the graphical user interface. Qt provides significant features for user interface, XML, and network editor code. Other languages including Java are used for secondary components such as stand-alone data processors called from the main program.

Component Libraries Used

Important components that were mentioned include the following. Some of these products have commercial licenses and incur costs that are recovered through RiverWare license costs.

- Qt (https://www.qt.io/) – user interface and other functionality (commercial license but also has GPL license option)
- CPLEX (https://www.ibm.com/analytics/cplex-optimizer) – optimization library (commercial license)
- QWT (https://qwt.sourceforge.io/) – plotting library (open source)
- Oracle Cloud Infrastructure (https://cloud.oracle.com/cloud-infrastructure) – Oracle database (commercial)
- Army Corps HEC-DSS (https://www.hec.usace.army.mil/software/hec-dss/) – integration with Army Corps platform (public domain) – this uses Fortran, wrapped by C/C++, wrapped by Java
- Geospatial Data Abstraction Library (GDAL, https://www.gdal.org/) – geographic information system (open source)
- Reprise (http://www.reprisesoftware.com/index.php) – software license management (commercial)
- FFMPEG (https://www.ffmpeg.org/) – audio and video streaming (open source)
- Other components are listed in the RiverWare ***About*** dialog, which provides required software component citations.

License and Open Source Considerations

RiverWare license information is available at http://www.riverware.org/riverware/LicensingRW/index.html. The RiverWare product is not currently open source and there are no plans to make it open source. Some of the important funding sponsors have traditionally not wanted the code to be open source for the following reasons:

- sponsors have paid for research and development that is considered proprietary
- sponsors want to know who uses the software
- the software is used in legal negotiations such as compact negotiations
- don't want forks of the code

Supported Operating Systems

The following is information about supported operating systems:

- RiverWare is currently supported only on Windows
- the original software was on Solaris (Unix) but migrated to Windows in 1999 and Windows and Solaris were both supported for about 10 years
- considered using Linux for more computing power but when Windows began supporting 64-bit architecture the pressure to support Linux was removed
- the code is generally platform neutral and code be ported to Linux if necessary
- have had conversations to provide a "headless" (no user interface) version, for example to run RiverWare as a component in the National Weather Service's Community Hydrologic Prediction System (CHPS)
- all funding agencies are using Windows

Development Environment

The development environment uses:

- Microsoft Visual Studio 2013 for C++
- Java adaptors use Eclipse integrated development environment (IDE)
    - these are stand-alone programs that interface with Java-based tools such as Army Corps Water Management System and Deltares FEWS system used by the National Weather Service
    - stand-alone programs are called to convert data into formats compatible with RiverWare
- Version control
    - Approximately 2 million lines of code
    - Git is used for version control
    - An internal university Office of Information and Technology server hosts the Git server and in the future a CADSWES server may be used

Software Testing

Software testing is implemented as follows:

- 45-50 regression tests are run using variations of models such as single reservoir up to full basin
- tests are run on the nightly software build with email notification if tests fail
- the testing framework is home-grown and is written in Perl/TK and is approximately 15 years old
- each developer does their own testing during development but there does not appear to be a formal unit testing (for example https://docs.microsoft.com/en-us/visualstudio/test/writing-unit-tests-for-c-cpp?view=vs-2017)
- user interface testing is difficult
    - the team did try to use automated testing when the Galaxy product was used for the user interface but it required too much maintenance
    - rely on in-house testing of user interface prior to release, which provides an approximately 6-month window between releases

Software Performance Optimization

- the nightly testing procedure also captures run time so that performance can be improved over time
- once a year or more often, have performance improvement effort, driven by users
    - have seen large improvement in performance over time
    - previously used Rational/Purify tool to evaluate performance but now use Microsoft Visual Studio (note that Rational/Purify was used in early versions of CDSS software but is no longer used because modern development environments typically provide built-in tools)

Development Team

The development team consists of:

- 9 technical people work on RiverWare, 8.5 FTEs.
- Would like to hire two more.
- Hiring is difficult, hard to find C++ programmers and moving to C-# would have impacts.
- Staff efforts are distributed among many activities, some not software related such as training and user support. Also some time is spent on web pages, updating hardware and software of developers, providing licenses, developing and maintaining download and installation procedures, etc. The RiverWare enterprise is fairly large since there are hundreds of users and much new development each year.
- Time is generally distributed as follows over the long term:
    - 40% on R&D of the software
    - 20% on software maintenance
    - 30% on user support and basin specific tasks (build models or help people build models, analyze and improve performance, review models, general user support)
    - 7% on training and developing training classes
    - 3% on managing licenses
- Software R&D: Of the time that is spent specifically on software R&D, there are many efforts beyond computation and GUI. Also, difficult to separate out GUI because most functionality has a GUI component. These things change every year, but below are recent things that have been worked on:
    - Computation-related: extension of RPL language (for expression of rules, objectives and constraints), development of new objects and new physical process methods, enhancement of water accounting and water rights algorithms. Last e.g., year we added new 5-minute and 15-minute timesteps.
    - Develop and maintain adapters that integrate RiverWare into larger systems including USACE CWMS and Deltares FEWS
    - Develop and maintain ancillary software such as RiverSMART for large studies and RiverWISE for scenario exploration by stakeholders
    - Develop and enhance output devices: plotting, auto-generated model reports, teacups and other animated displays that can be exported, etc.
    - Maintain and extend data management interfaces, direct connections to various databases, DMI performance

    - Usability features like model and ruleset comparison tools that facilitate group development of models
    - Scripting that allows setup, configuration and execution of many steps in model setup and executing.
    - Geospatial capabilities
- Software maintenance: The software maintenance effort should not be underestimated. For RiverWare it includes updating the software to keep current with operating systems, versions of 3rd party libraries (Qt, CPLEX and about a dozen others) and compilers; releasing new versions of RiverWare including testing for releases, generating release notes, maintaining release areas, release scripts and supporting tools; maintaining and constantly updating regression test procedures, maintaining issue-tracking software and web pages, maintaining download and install capabilities, licensing scripts, and development protocols; and fixing reported issues. A current effort is to generate html online help (moving from old pdf system).
- Software developers work on the user interface and water resources staff develop analysis first.
- No students directly work on code (all work is input to other programmers).
- RiverWare is more complicated than StateMod in some ways given optimization, multiple run module (ensembles), etc.
- Sponsors pay for enhancements, which fund positions.
- The longevity of people is important given the complexity of the product.
- The "brain trust" is the full 8 people and it would be difficult to do without any person.
- The person responsible for user interface development, with 20 years of experience, passed away recently and it was a major blow to the team.
- In addition to the core product, staff focus on adaptors written in different languages, which requires various skills.