Astronomical Data Analysis Software and Systems IX
ASP Conference Series, Vol. 216, 2000
N. Manset, C. Veillet, and D. Crabtree, eds.

The MERLIN Correlator Software Project

E. Troup, S. T. Garrington, N. P. F. McKay, R. G. Noble

Jodrell Bank Observatory, University of Manchester, Macclesfield, SK11 9DL, UK

Abstract. The MERLIN[1] array is being upgraded with new correlator hardware and improved monitoring, operating and control software. The aims of the upgrade are to provide easy to use display and monitoring of all components of MERLIN, including the astronomical data, to improve reliability and to assist the MERLIN controllers in data quality monitoring. The project design is based around object-oriented techniques and is being coded in C++. The project is being developed using elements of the JIVE[2] correlator software, also designed at Jodrell Bank.

1. Introduction

The MERLIN array is a network of radio telescopes connected by microwave links into a real-time interferometer. The radio telescopes are located in England and controlled from Jodrell Bank Observatory via land lines. A small team of operations staff schedule the running of the array while controllers monitor the operation and safety of the telescopes while the system is in use. The observing system is under the overall control of a computer attached to the correlator and this computer is responsible for sequencing the observations, configuring the correlator and accumulating the correlated data. The project aims to improve the reliability and efficiency of MERLIN operations by upgrading the real-time correlator control computer and providing new software for aiding astronomical data-quality and array subsystem monitoring. The software is being designed using object-oriented methods and is written in C++. Software analysis has proceeded using the methodology described in Rumbaugh et al. (1991).

Wherever possible, the MERLIN upgrade project intends to re-use software designed for the European VLBI Network[3] MkIV correlator.

2. Aims

The aims of the project include providing the following features:

[1]MERLIN, The Multi-Element Radio Linked Interferometer Network is operated by the University of Manchester on behalf of PPARC

[2]`http://www.nfra.nl/~jive/`

[3]`http://www.nfra.nl/jive/evn/evn.html`

- Correlator upgrade: Install a new real-time control processor for the correlator. Install a workstation to configure, control and monitor the system. Re-use or adapt JIVE/EVN correlator software where possible.
- Array Monitoring: A graphical display of the array including telescopes, modems, L-band and microwave links.
- Data Analysis: Scripts for the operators to automate the quality control of data from calibrator scans. Check and report on telescope gain, polarization performance, closure phase, etc.
- Data Display: Display correlation (amplitude and phase) for each baseline. Display lag data in real time. Scrolling display of a single frequency channel or the average of several channels.
- Data Processing and Archive: Operators to scrutinize the majority of data within a few days. Scripts to automate processing e.g. calibration, phase-referencing and imaging. Operators will also be responsible for archiving the data and maintaining the database of completed observations.
- Scheduling: Flexible scheduling so that the controllers can reschedule important scans if weather or equipment cause problems. The operators need a simple view of the current schedule with highlighting for high priority scans.

3. Implementation

The development of the system has begun using software and software models developed for the JIVE correlator, as well as MERLIN-specific code. A large amount of work is still to be done, but the availability of well-tested class libraries has sped up development and prototyping. An example of an application developed for MERLIN is shown in Figure 1. This display is the output from a test correlator and was built using shared memory classes derived from JIVE classes and the Qt and Qwt class libraries.

The class libraries being used to implement the MERLIN software include the following:

Shmem_Shared Using the Shmem_Shared class as a base, any derived classes will reside in shared memory. Shared memory classes can be used for fast, efficient transfer of bulk data, such as correlator output data or monitor data between Jobs.

Message The Message class is designed to implement a machine- and transport-independent interprocess communication class. This is used for such things as sending commands or the response to commands between processes.

Threads The Threads classes are used to implement concurrent streams of control within a process.

State_Machine The State_Machine class provides a framework for building a state machine within a process. It can be used in conjunction with the Threads classes to implement the states and transitions of a software dynamic model.

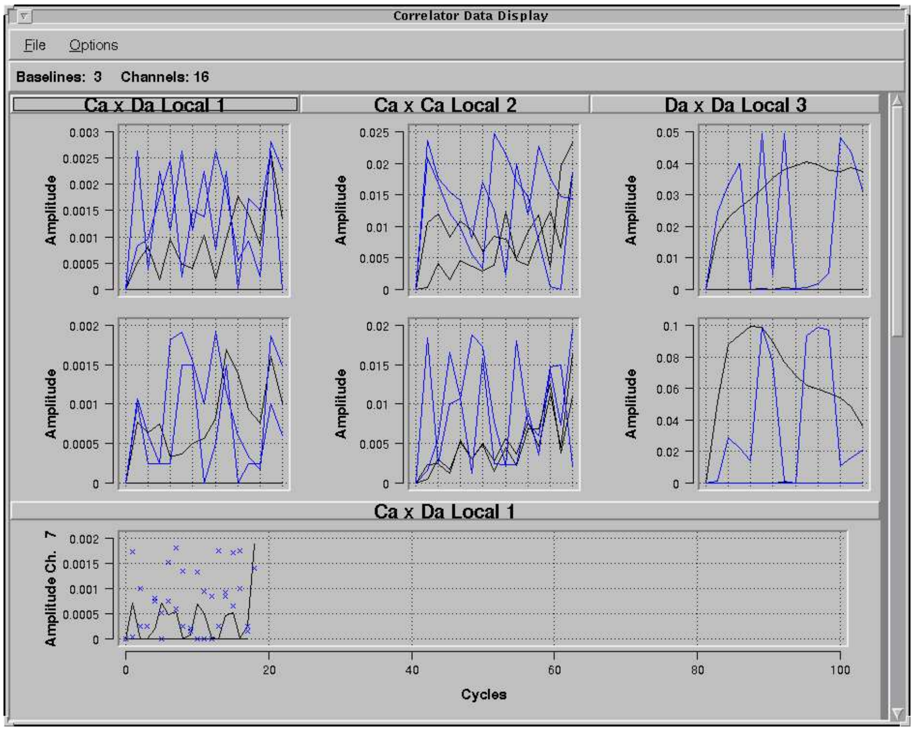

Figure 1. Real-time correlator data display.

Utility A number of utility classes including a hash-table class and a circular buffer class.

Qt Free Edition Qt is a user-interface C++ class library produced by TrollTech[4]. It includes a wide range of widgets and is designed for ease of use and customization.

Qwt Qwt[5] is a widget set based on Qt and provides x-y graph displays with both log and linear scales and includes scrollable data plots that can be used for real-time data.

References

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. 1991, Object-Oriented Modeling and Design (New Jersey: Prentice Hall)

---

[4] `http://www.troll.no`

[5] `ftp://metalab.unc.edu/pub/Linux/libs/X/c++libs/qt/qt-0.2.tar.gz`