S91-82
ABS. ONLY

.175081

N94-22529

The Data Acquisition System for the AAO 2-degree Field Project

K. Shortridge, T.J. Farrell, J. Bailey (AAO)

The Anglo-Australian Observatory (AAO) is building a system that will provide a two-degree field of view at prime focus. A robot positioner will be used to locate up to 400 optical fibres at pre-determined positions in this field. While observations are being made using one set of 400 fibres, the robot will be positioning a second set of fibres in a background field that can be moved in to replace the first when the telescope is moved to a new position. The fibres feed two spectrographs each with a 1024 square CCD detector. The software system being produced to control this involves Vaxes for overall control and data recording, UNIX workstations for fibre configuration calculations and on-line data reduction, and VME systems running VxWorks for real-time control of critical parts such as the positioner robot. The system has to be able to interact with the observatory's present data acquisition systems, which use the ADAM system. As yet, the real-time parts of ADAM have not been ported to Unix, and so we are having to produce a smaller-scale system that is similar but inherently distributed (which ADAM is not). We are using this system as a testbed for ideas that we hope may eventually influence an ADAM II system.

The system we are producing is based on a message system that is designed to be able to handle inter-process and inter-processor messages of any length, efficiently, and without ever requiring a task to block (ie be unresponsive to 'cancel' messages, enquiry messages), other than when deliberately waiting for external input — all of which will be through such messages. The essential requirement here is that a message 'send' operation should never be able to block. The messages will be hierarchical, self-defining, machine-independent data structures. This allows us to provide very simple monitoring of messages for diagnostic purposes, and allows general purpose interface programs to be written without needing to share precise byte by byte message format definitions.

Programs in this system have interfaces defined simply in terms of named actions and their parameters. Real-time control programs are required to be able to handle a number of such actions concurrently; data reduction programs will normally only need to handle one action at a time ('process an image', 'display a spectrum', etc).