A distributed, real-time data monitoring system as ground support equipment for balloon-borne astronomy experiments

C. M. Hubert Chen, Wayne H. Baumgartner, Walter R. Cook, Andrew J. Davis and Fiona A. Harrison
California Institute of Technology, Mail Code 292-17, Pasadena, CA 91125, USA

hubert@srl.caltech.edu

Context
We developed a real-time data-monitoring software suite for the High Energy Focusing Telescope (HEFT). HEFT, one of the first hard X-ray focusing telescopes (20–70 keV), was launched on a balloon-borne platform from southwest USA in 2005. This software suite was our ground-station equipment for monitoring the focal-plane instruments during on-site calibration, pre-launch practice drills, and an observation flight of 25 hours.

Features / Design objectives
• Distributed and scalable: servers + clients.
• Mostly platform-independent: Java (+ 1 perl client), simple UDP datagrams.
• Task-specific servers provide data redundancy.
• Allows for both real-time data input and playback of saved data.

Server-client architecture
• Multiple servers performing dedicated tasks guarantee the integrity and redundancy of logged data, and reduce server load.
• UDP multicasting of photon & sensor data makes possible an unlimited number of concurrent display clients, without increasing server load.
• Separation of data processing and display makes code development modular.

Features / Design objectives
• Distributed and scalable: servers + clients.
• Mostly platform-independent: Java (+ 1 perl client), simple UDP datagrams.
• Task-specific servers provide data redundancy.
• Allows for both real-time data input and playback of saved data.

Server-client architecture
• Multiple servers performing dedicated tasks guarantee the integrity and redundancy of logged data, and reduce server load.
• UDP multicasting of photon & sensor data makes possible an unlimited number of concurrent display clients, without increasing server load.
• Separation of data processing and display makes code development modular.

Display clients
Detector maps, in sky coordinates

- Rotated detector maps show X-ray images as seen from sky.
- Event selection in time and energy (wavelength) capable.

Detector maps, in detector coordinates

- Unified interface for displaying pixel-specific data: photons, test pulses, leakage current, baseline voltage, etc.
- Near-real-time update, interval adjustable.
- Event selection in time and energy (wavelength) capable.
- Resting pointer at a pixel displays pixel coordinates and value as tooltip.
- Clicking on a pixel or dragging across corners of a group of pixels displays light curve and spectrum in new window.

Housekeeping sensor readings

- Unified display of housekeeping sensor readings (of pressure, temperature and voltages), showing latest readings, 5-minute averages and standard deviations.
- Near-real-time update, interval adjustable.
- Clicking on a cell displays time-series in new window.
- Colour and (optional) audio alarms when a value goes out of expected range.

Time series, light curves and spectra

- Unified display of time series, light curves and spectra.
- Near-real-time update, interval adjustable.
- Event selection in time and energy (wavelength) capable.
- Zoom in by dragging pointer from top-left corner of zoom window to bottom-right; zoom out by the reverse.

Implementation detail
Data server: Data flow

- Multi-threading provides modularity, enables incremental development.
- Task-specific servers (logging, playback) reuse the same code, running only relevant threads.
- Events from input serial port drives execution, minimizes risk of data loss.

Data server: Data structures

- Data structures in both time and energy order allow efficient insertion, retrieval and user interface.
- Data from housekeeping sensors share the same data structure as pixel-specific data from detectors, reducing code complexity.

Display clients

- Multi-threading separates timed data request/input, graphical rendering, and user interface.
- Shared, common code for TCP/IP communication with data server for all Java clients, while graphical front ends vary.

Network interface

UDP multicast

- Latest science and housekeeping data require immediate mass distribution; occasional data loss is unimportant.
- Data server transmits data as UDP datagrams to designated IP multicast group address (and port) at 1 Hz; display clients ‘tune’ into the same group.
- Platform- and language-independent protocols make possible separate code development for server and client, and modular upgrade/rewrite.

TCP/IP unicast

- User-selected time series, light curves and spectra requires dedicated communication line.
- Display client sends data requests (data type: [photon/test pulse/vetoed/baseline]

Performance and testing

We ran the system daily over two months-long flight campaigns, including a 110-hr-long continuous calibration run (wired input directly from the focal plane) and a 25-hr-long balloon flight (input from radio telemetry with data dropouts). Changes were made iteratively over the campaigns until a final code freeze in view of the balloon launch. The software has since been functioning, meeting specifications.

Invitation to adopt

This system, and individual ideas of its implementation, can be adapted for use in future experiments requiring sophisticated real-time monitoring and data display. We welcome discussions of prospects for collaboration and code reuse.

Acknowledgements
This work is funded by the NASA Space Science “Supporting Research and Technology” (SR&T) programme under award number NAG5-5398.