

The operator GUI of the Cherenkov Telescope Array

I. Sadeh,

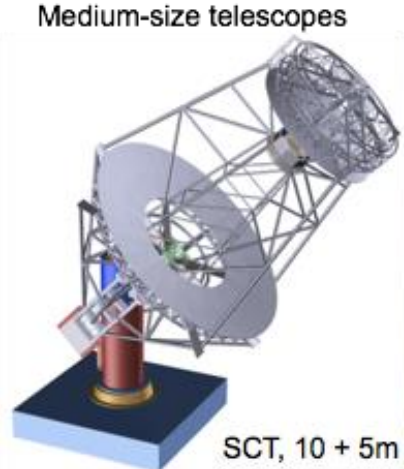
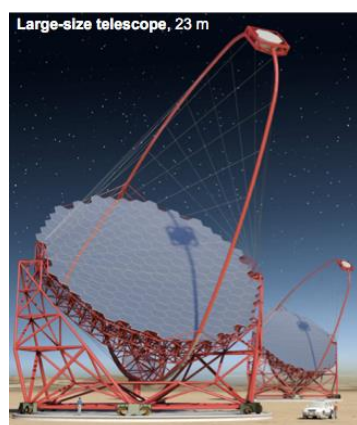
I. Oya, J. Schwarz, E. Pietriga & D. Dežman

For the CTA Consortium

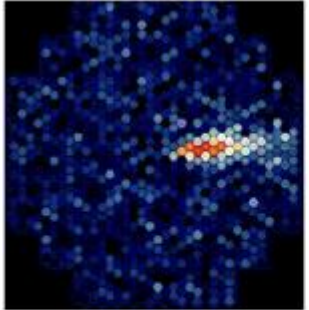
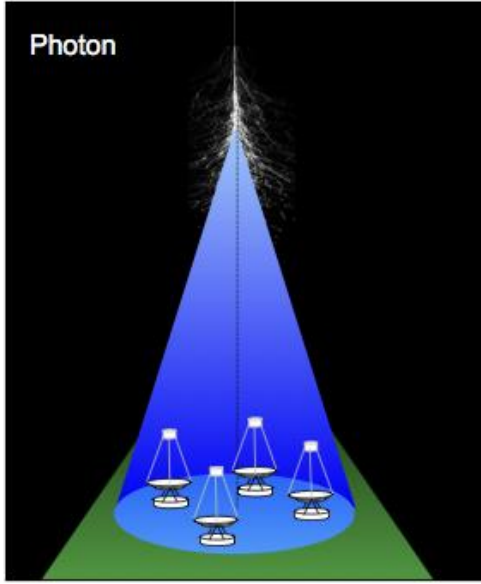
CTA: <https://portal.cta-observatory.org>

GUI prototype: <https://www-zeuthen.desy.de/~sadeh/>

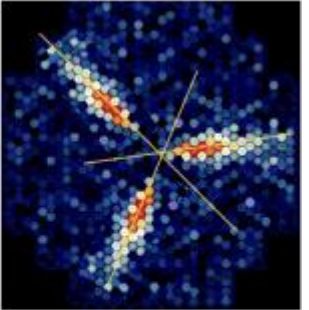
October 2017



the Cherenkov Telescope Array



Event in a single telescope



Images from 3 telescopes in the common camera plane

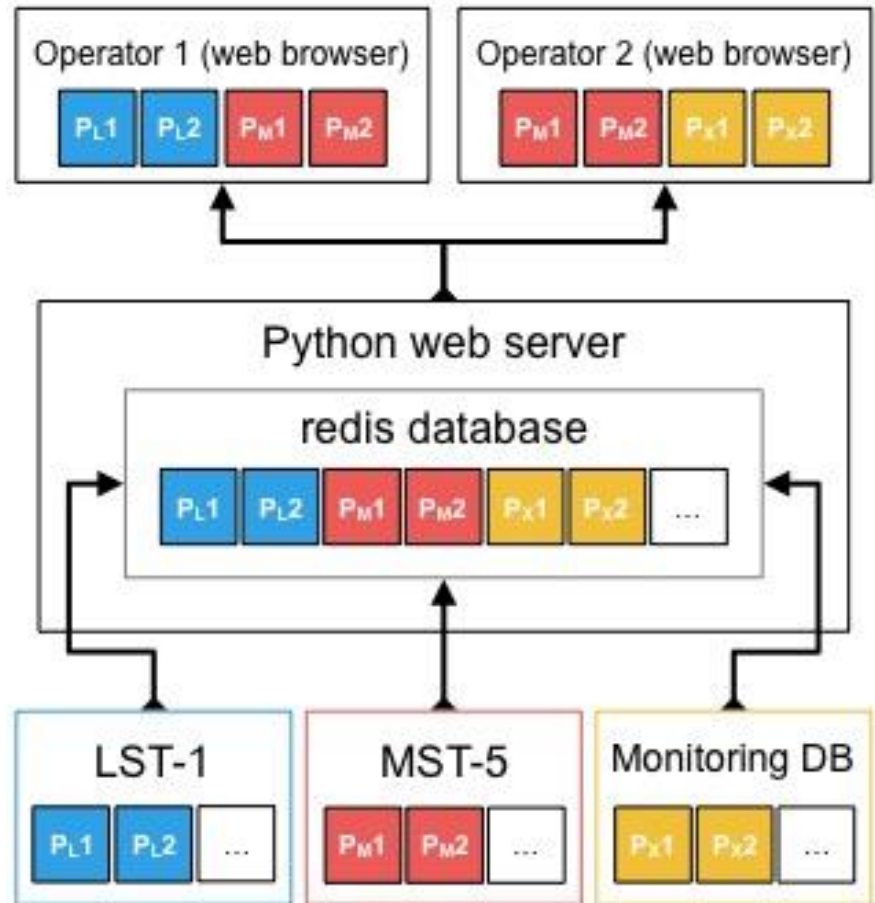


Designing the operator GUI

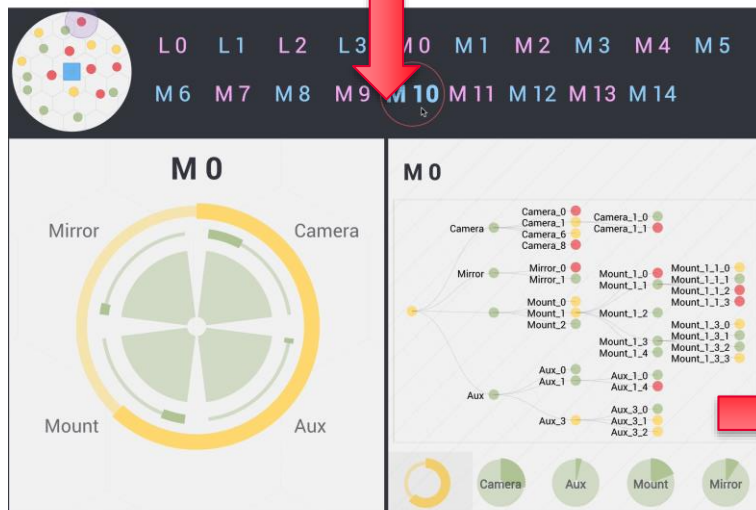
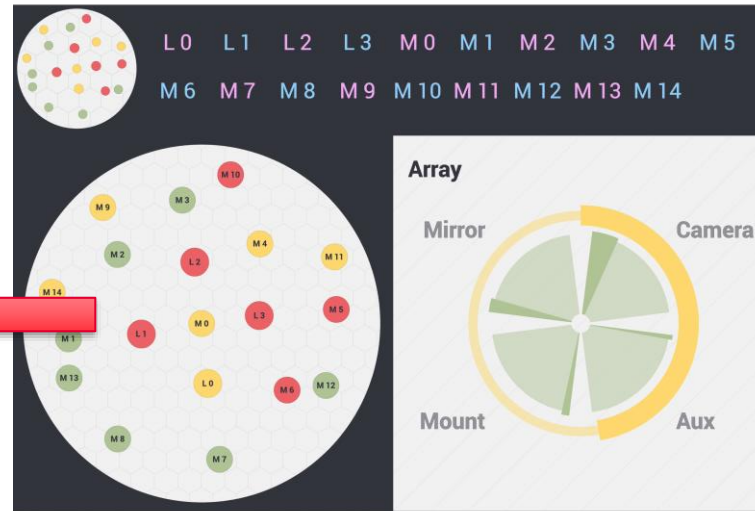
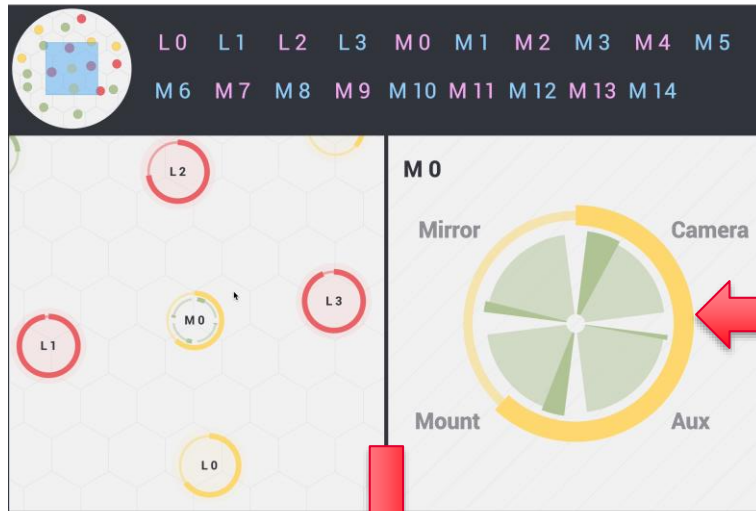
- We conduct participatory design workshops with human-computer interaction (HCI) collaborators and various teams from CTA.
- The following questions are used as design guidelines:
 - What should the GUI enable users to do?
 - How should the GUI be designed?
- Current prototype under development:
 - Server: Pyramid, a lightweight python web framework.
 - Data access / buffering: redis, mongodb, ACS.
 - Asynchronous communication between front- and back-end: socket.io web-sockets lib.
 - Front-end framework: - Polymer, a Material Design, Web Component framework by Google.
 - Data visualization: d3.js/dc.js(crossfilter) Javascript libraries using SVG, HTML5 & CSS for real-time interactive data visualization.

Information flow & integrated redis database

- Exposing monitoring data to users of the GUI.
 - Two users (Operators 1 and 2) request data from a combination of sources.
 - The data transmitted to the users are stored in a redis database, which is integrated with the Python Web server.
 - redis is filled by directly accessing lower-level elements of the system. In this example, these are two telescopes (LST-1 and MST-5), and the monitoring database.

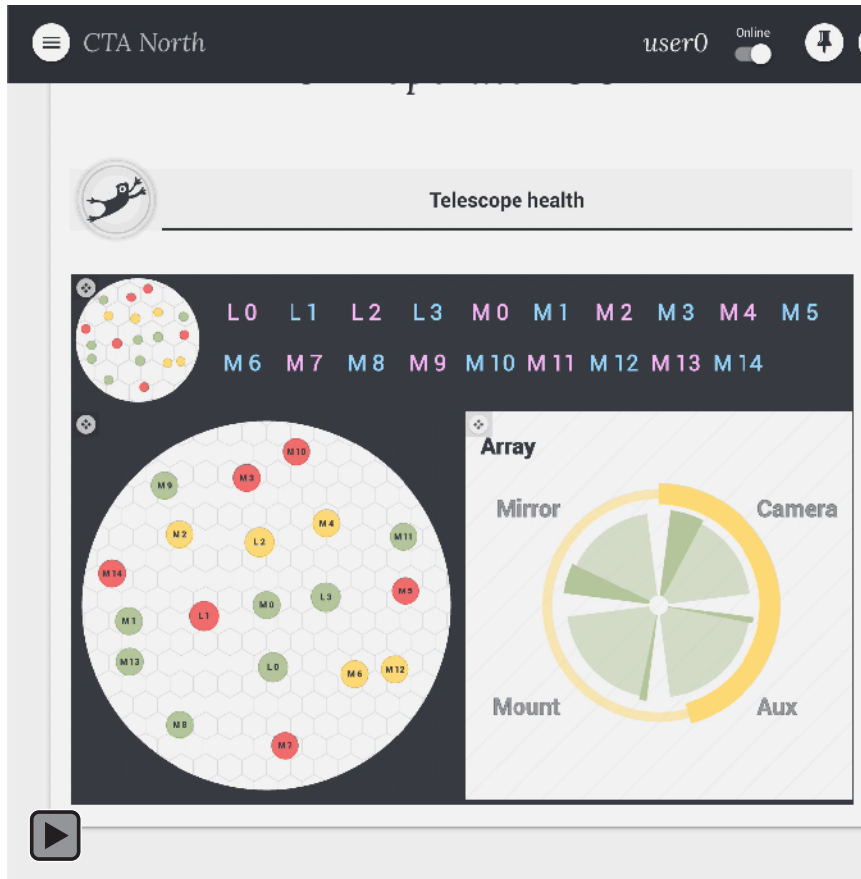


Monitoring panel with semantic zoom

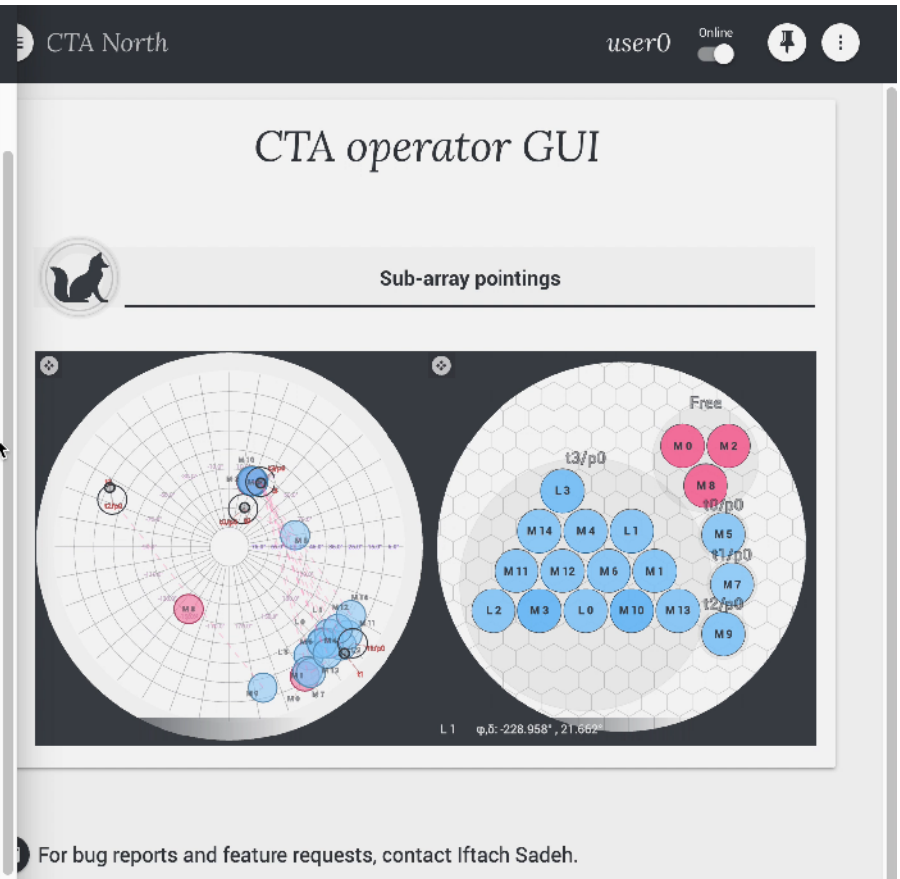


Example panels with sync.

Telescope status monitoring



Telescope pointing positions



More movies available at: <https://www-zeuthen.desy.de/~sadeh/>