Telemetry

Operations UI

HISTORY

- The current OpsUi was preceded by the RcsGUI.
- RcsGUI used a series of client pull requests (polling) to extract status from the RCS control-server on port 9110.
- Only services within the RCS were available.
- Only continuously changing sources of data were obtainable, i.e. no event-driven feeds.
- Initially created to monitor a very limited set of data from RCS.
- Developed over a number of years on an ad-hoc basis.

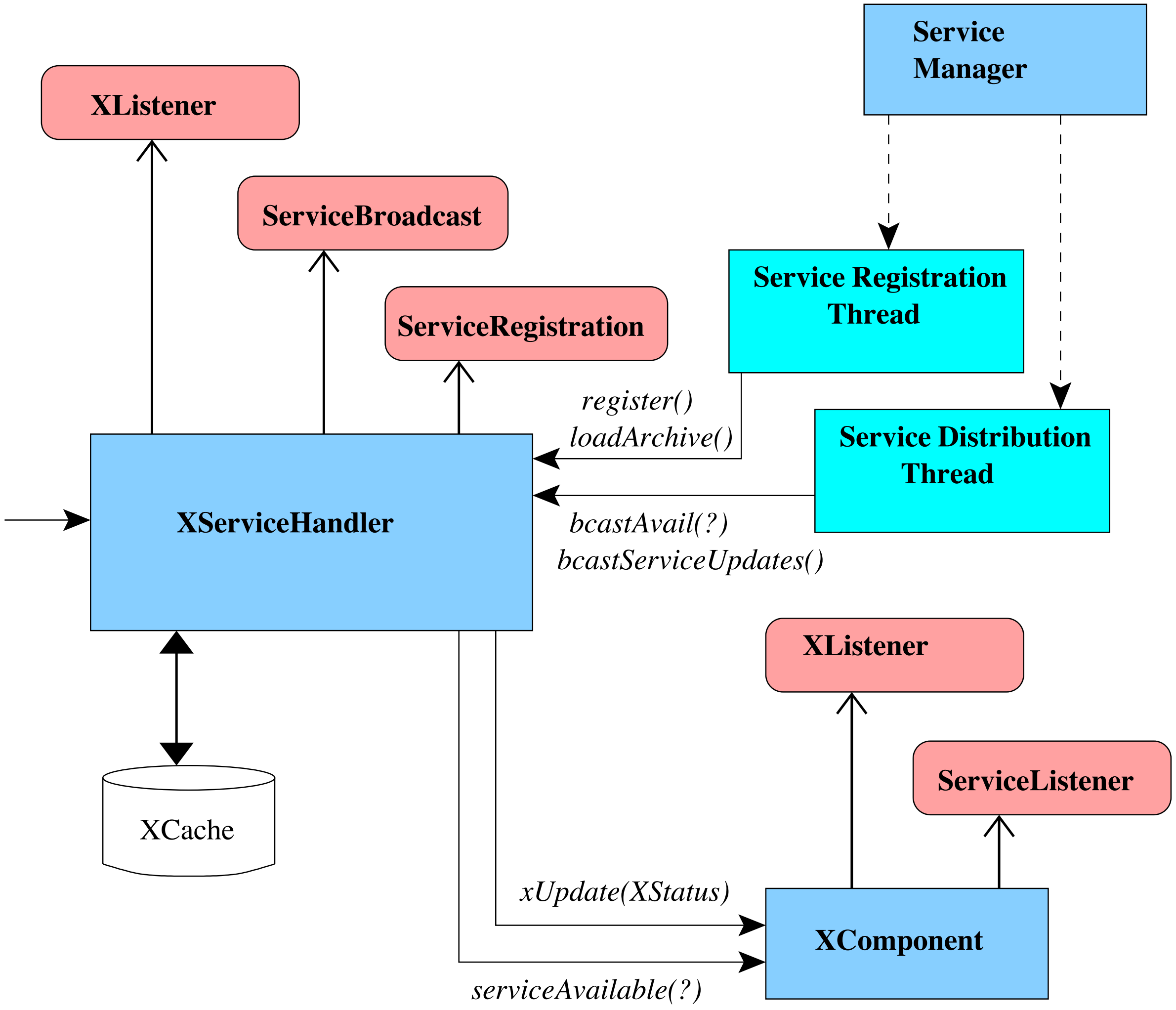
OPS UI

- Goal was to have access to all available feeds, both continuous and event-driven from any system capable of being monitored.
- Potential systems:-
  - RCS
  - Instruments
  - Environment
  - Scheduler
  - Synoptic Models
  - OSS
  - TEA
  - NSO
  - Skycams

Design – data collection

- Each feed source is designated as a service and has a specific *ServiceProvider (*SP*).*
- Different classes for each feed.
    - E.g.
        - ngat.opsgui.services.TelescopeStatusHandlerService
        - Ngat.opsgui.services.ReactiveSystemHandlerService
- Managed by the *ServiceManager* (SM).
- Each provider is a listener for the feed source it is attached to.
    - *E.g. TelescopeStatusHandlerService* implements *ngat.tcm.TelescopeStatusUpdateListener*.
- Remote source (or gateway) updates SP which stores the received status in a queue.
- SM runs a registration thread and distribution thread for each service.
- Registration thread regularly re-connects to the provider (or gateway) incase the registration is lost or source restarts.
- Distribution thread broadcasts the status from the queue to any connected handlers.

DATA HANDLING



Design - UI

- The OpsUI screen is made up of several areas.
- Side-bar
  - Always visible
  - Contains sub-components which show summary status for different services.
- Top-bar
  - Always visible
  - Contains sub-components which show summary information.
- Perspectives area
  - A number of tab panels (perspectives) loosely associated with the various SPs.
  - Perspectives are themselves tab panes and contain data in separate tabs associated with different aspects of the services they display.
- Menu bar
  - Fixed menus and perspective-dependant menus.

Available Services

- INSTR
  - InstrumentStatusHandlerService
    - implements ngat.icm.InstrumentStatusUpdateListener
- METEO
  - MeteorologyStatusHandlerService
    - implements ngat.ems.MeteorologyStatusUpdateListener
- OPS
  - OperationsEventHandlerService
    - implementsngat.rcs.ops.OperationsEventListener
- PHASE2
  - Phase2CacheService
- ERS
  - ReactiveSystemUpdateHandlerService
    - implements ngat.rcs.ers.ReactiveSystemsUpdateListener
- SCHED
  - SchedulingHandlerService
    - implements ngat.sms.SchedulingStatusUpdateListener
- SKY
  - SkyModelHandlerService
    - implements ngat.ems.SkyModelUpdateListener
- TASK
  - TaskLifecycleEventHandlerService
    - implements ngat.rcs.tms.taskLifeCycleEventListener
- TEL
  - TelescopeStatusHandlerService
    - implements ngat.tcm.TelescopeStatusUpdateListener

Available Perspectives

- Astrometry
- Instruments
- Meteorology
- Operations
- Phase2
- Reactive
- Scheduling
- Services
- Tracking
- Task

STRUCTURE

- Perspectives display data associated with one or more telemetry feeds. They have several tabs to display different types of information.
- Perspectives are grouped into Displays. A display can have one or more perspectives, selected using the tabs on the right. (these are meant to have icons rather than letters, a job for someone with graphic design capabilities.
- A collection of displays is a Layout. These can be setup at startup – in principle this should be configurable per user but needs a little more work.
- One display has the sidebar and topbar. If only a single display in the layout then it has them.
- Individual perspectives can be torn off the display and popup in their own standalone display. Clicking the close or minimize icon will pop them back where they came from.
- Some individual tab panels within a perspective can be torn off – this currently only works with the Scheduling perspective.

Perspectives

- Perspectives act as handlers for the various service broadcasts.
  - e.g.*TrackingPerspective* receives updates from both the Telescope and Scheduling services.
- They then distribute the data to their sub-components – i.e. the sub-panels. These in turn update individual components – datafields, status indicators, graphs.
- Each perspective has menus associated with it.
- When a perspective is selected, the main menu bar contains those menus in addition to those which are always present.