Design Patterns for the SKA Control System

ICALEPCS 2021

Sonja Vrcic
Control System Architect
SKA Observatory

Overview of the talk

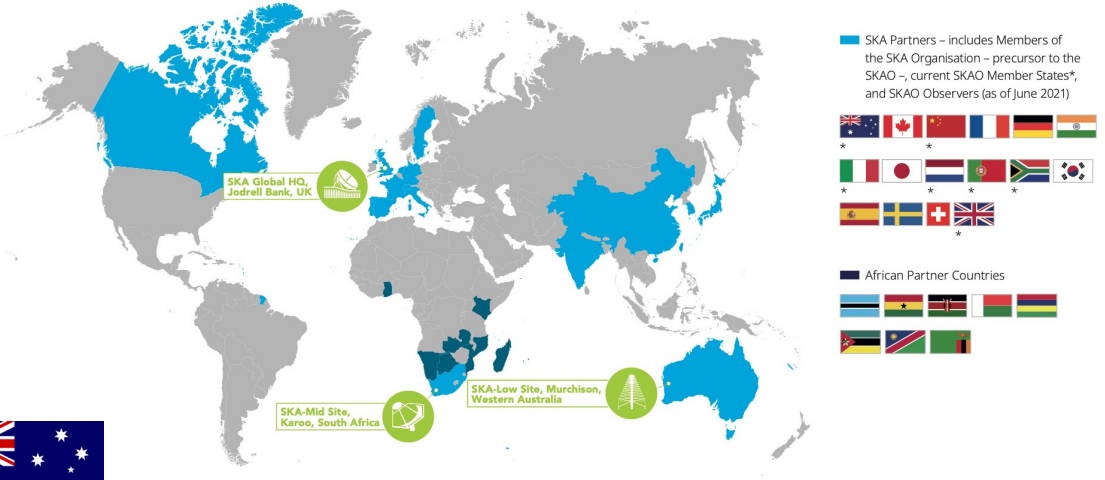
- Square Kilometre Array – a brief overview
- SKA Telescope Control System
  - What do we need to monitor and control
  - Requirements, challenges
  - Design patterns

SKA Observatory

1-2-3
- one observatory,
- two radio-telescopes,
- on three continents.

- Inter-governmental organization

https://www.skatelescope.org/



- Design was developed during the pre-construction.
- Successfully completed the Critical Design Review.
- **Construction started on 1. July 2021.**
- Contracts are being awarded.
- Team is growing.

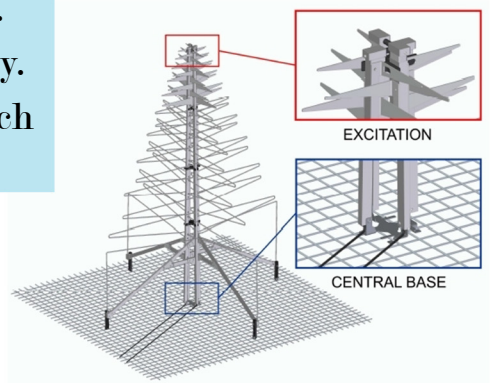
SKA Science Goals

https://www.skatelescope.org/science/

- Galaxy evolution, cosmology and dark energy
- Strong-fields tests of gravity using pulsars and black holes
- The origin and evolution of cosmic magnetism
- Probing the cosmic down
- The cradle of life
- Flexible design to enable exploration of unknown…

SKA – Square Kilometre Array

- The goal is to (eventually) build a radio-telescope with collecting area of 1km^2.
- Instead of building a single gigantic dish, use a technique called interferometry.
- The waves are superimposed to cause the phenomenon called interference, which is used to extract information.

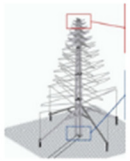


★**Global HQ**
- Jodrell Bank Observatory, Manchester, UK

★**Low Frequency Array Telescope**
- Murchison region, Western Australia
- Observing range: 50 - 350 MHz
- 131,072 (512 x 256) log periodic antennas

★**Mid Frequency Array Telescope**
- Karoo region, South Africa
- Observing range 350MHz - 15GHz
- ~200 x 15m diameter dishes

Radio-Telescope - an overview



electromagnetic vawes

Receptors

Collect and digitise band of interest

Mid Dishes:
servo motors
Low and Mid:
analog and digital hardware
(PLC / FPGA / CPU)

packetised and timestamped data samples

Digital Signal Processing

Real-time processing
correlation
beam-forming

digital hardware
COTS and custom
FPGA / CPU / GPU
network switches

visibilities and beamformed data

Data Processing

Near real-time processing
pulsar search & timing
calibration
creating data-sets

high-performance
computing platform

FPGA / CPU / GPU
COTS and custom

data sets corrections pulsar candidates

Batch Processing

Batch processing
imaging

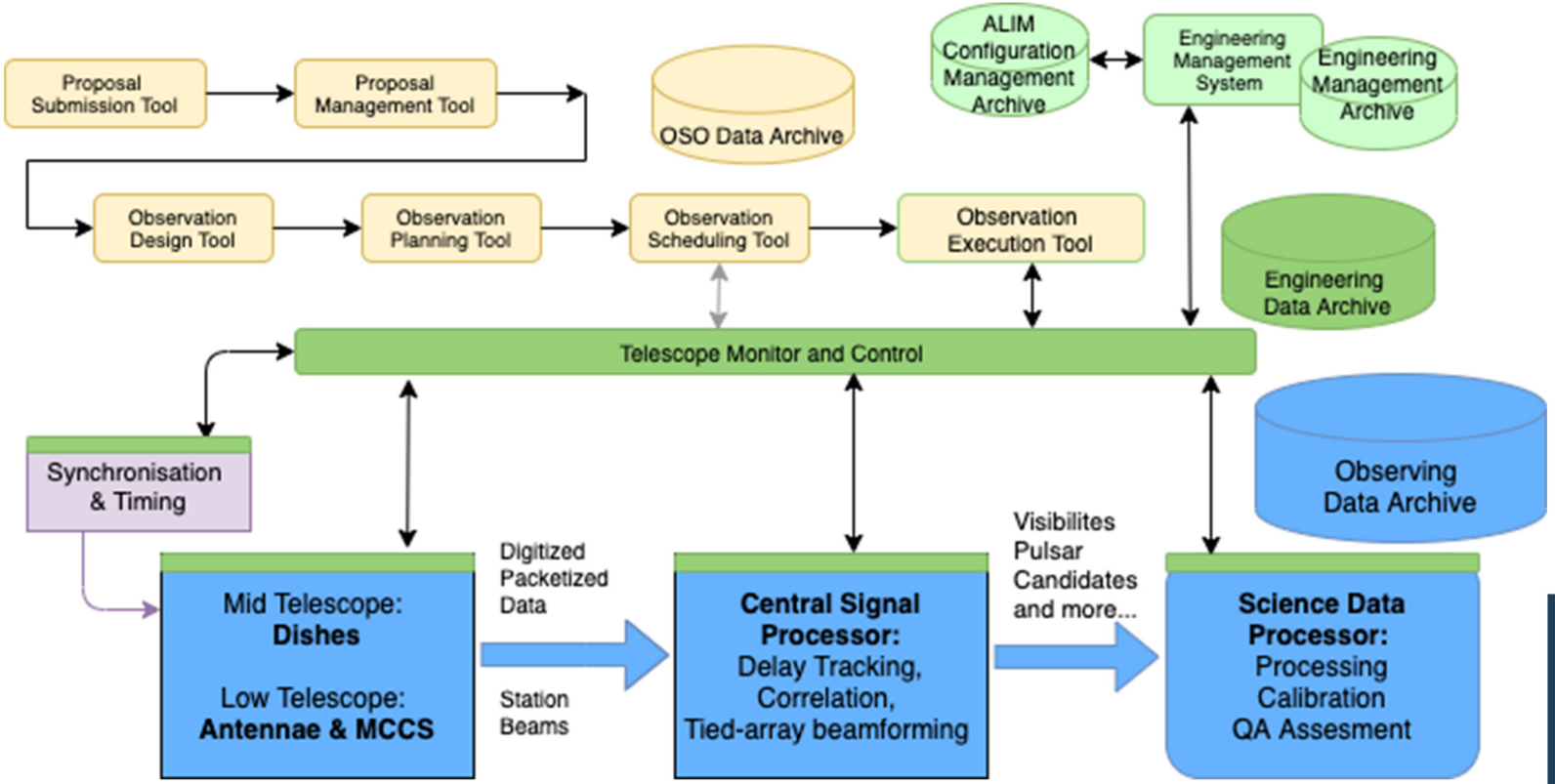
high-performance
computing platform

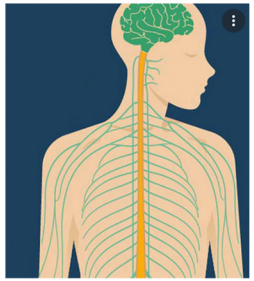
image cubes pulsar data

Infrastructure: facilities, roads, lending strips, power, water, networks, masers, IT...

Data Archives: Observing Data, Scientific Data, Engineering Data, Project Data...

SKA Telescope - software overview





SKA Telescope Control System Requirements

Equipment:

- Monitoring (report status, including faults and errors).
- Archiving (periodic reporting).
- API to trigger state/mode changes (command).
- Support for hardware / software / firmware updates.
- Support for debugging, testing and maintenance.

These requirements apply for:

- Telescope
- Sub-systems
- Individual components

Observing:

- Each telescope supports multiple observing modes - configure a telescope for the desired observing mode.
- Point the receptors and beams; track sources.
- Start and stop data processing.
- Store data products (data sets).
- Subdivide the array (receivers and processing resources), operate each sub-array independently (in terms of observing band & mode, start and stop).

SKA Telescope Control System
Physical vs Functional View

- ➔In an interferometer, the functional view does not always directly map to the physical view.
- ➔Requirement to operate each sub-array as an independent telescope further complicates ‘mapping’ of the functionality to physical equipment.
- ➔Control System provides two views:
  - Physical (equipment and components).
  - Functional (subarrays, capabilities).

TANGO Controls Framework

- During the design phase a decision was made to use TANGO Controls as a base for implementation of the Telescope Control System.

  https://www.tango-controls.org/

  - TANGO Controls is an open-source device-oriented toolkit for controlling any kind of software and hardware.
  - TANGO Controls can be used to build Distributed Control Systems and SCADA (Supervisory Control and Data Acquisition) Systems.

- From the design point of view the key concepts are:
  - TANGO Device - a software component (instantiation of a TANGO Device Class). Each TANGO Device models device, software component, or sub-system.
  - Tango Device Server – an execution environment for one or more TANGO devices.

SKA Telescope Control System Architecture

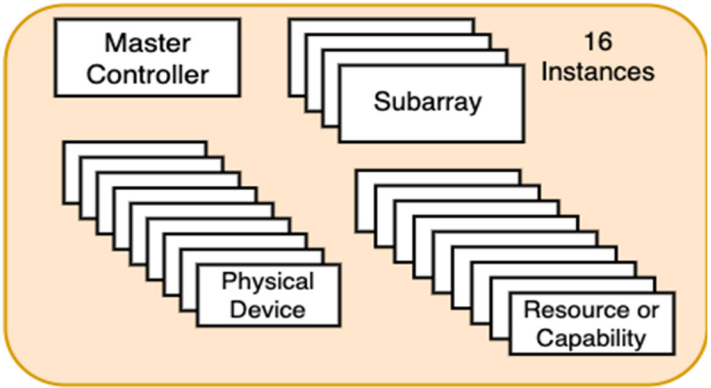
- Key telescope sub-systems consist of hundreds, if not thousands, of components and are organized hierarchically.
- Each ‘level’ performs aggregation and reports overall status of all subordinate components.
- Each level provides TANGO API.
- Commands are used to pass the observing mode configuration from top to bottom.
- At each ‘level’ the higher-level parameters are translated into detailed configuration of the subordinate components.

Sub-systems that implement sub-arraying implement this design pattern:

- Master Controller
- Physical devices
- Subarray
- Capability

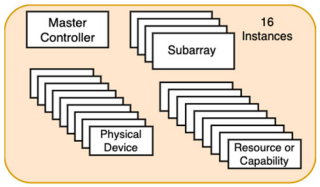
\+

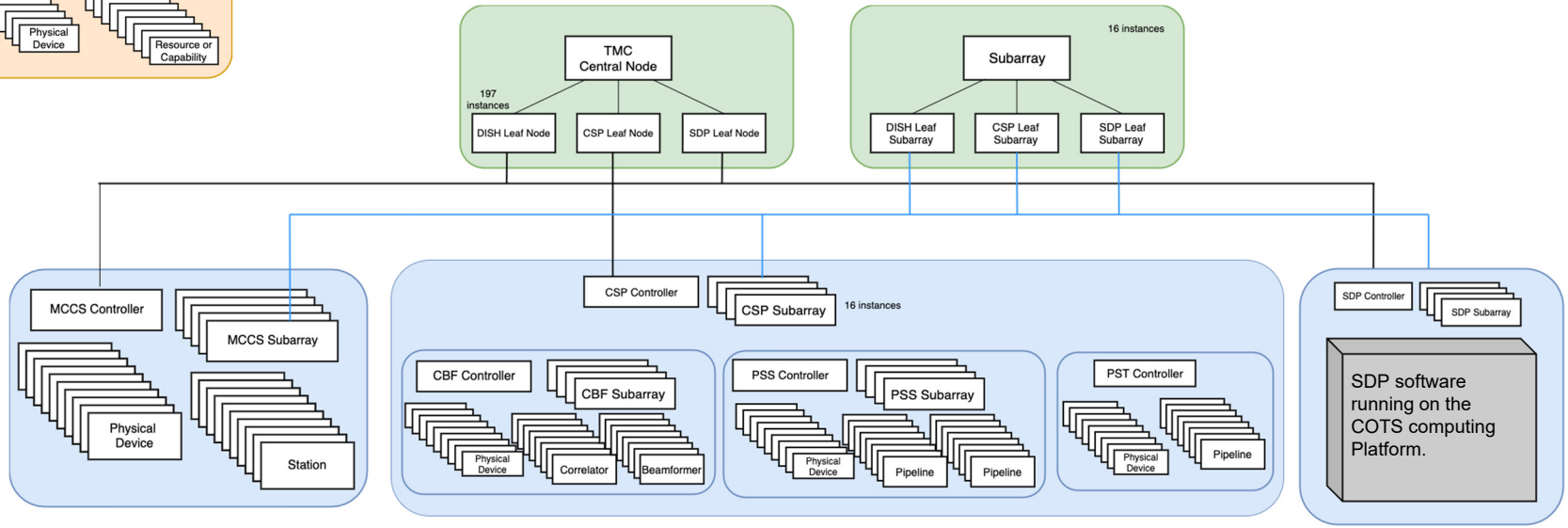
Standard set of state and mode indicators



SKA Telescope Control System
A simplified overview

Design pattern



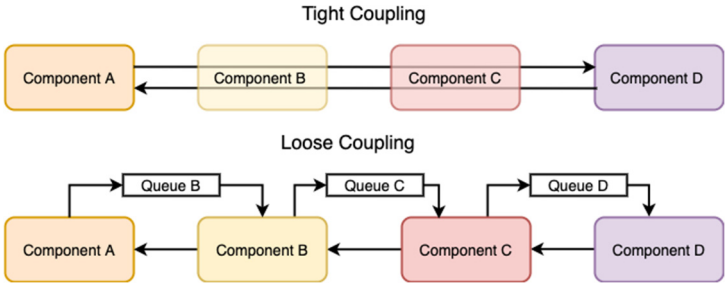


SKA Telescope Control System - Design Pattern
Decoupling TANGO from the "business logic"

- ➔Hierarchical organization
- ➔"Deep adoption of TANGO" - TANGO API provided at all levels of hierarchy:
  - Top-level view (overall status of the telescope).
  - All low level components (equipment, software components).
  - And also for the 'middle' i.e. all the sub-systems
- ➔Unwanted consequences:
  - The framework (TANGO) becomes entangled into every aspect of the control system logic; *overly dependent on* TANGO.
  - Confusion regarding the TANGO Device (software model) and the device being modeld.
- ➔Solution: Decouple the TANGO API (layer) from the 'business logic'.
  - TANGO Device provides API (attributes, commands, alarms, events).
  - Another class, so called Component Manager implements the logic required for monitoring, configuration and control.

SKA Telescope CS - Design Patterns
Loose Coupling of Components

- Loose coupling of components achieved using the following techniques:
  - Asynchronous communications.
  - Components implement input queue (optional).
  - Use of JSON (hi-level messaging, weak typing).



- Overview of the command implementation:
  - Before issuing a command a client registers to receive events.
  - When a command is received, the server adds the commands to the queue (FIFO).
  - Control is immediately returned to the caller.
  - A worker thread, when idle, removes the command form the queue and executes.

SKA Telescope Control System Design Patterns

Summary

- Hierarchical organization
- Decouple physical and functional view
- Distributed control of process logic *
- Set of SKA Base Classes (standard set of state/mode indicators, commands and more)
- TANGO Control framework
  - Provides physical connection via a mediator (CORBA) *
- TANGO API provided at all levels of hierarchy
- Decouple TANGO API from the ”system logic”
- Use JSON to pass configuration messages
  - Data-centric and self-contained messages *
  - Simple common types in data model *
- Asynchronous communications *

* Promotes loose coupling of components

Thank you !

Sonja Vrcic
Control System Architect
SKA Observatory