Status of the Computing TF and next steps

Kjeld v.d. Schaaf

Netherlands Organisation for Scientific Research

- TF splitted into separate Computing and Calibration & Imaging task forces
- Many thanks to Tim Cornwell

- Whitepaper v0.5 input for SKA engineering overview
- Rather complete overview of
  - requirements
  - the issues software related to SKA software
- Introduction to development approach

FIGURE 1. ALMA software: as categorized by Kemball and Cornwell (2004)

Subsystem	Purpose
Process	
Management	*Management of computing development*
Science software requirements	*Development of scientific requirements, scientific oversight, testing, etc.*
Analysis and design	*High level analysis and design of the entire ALMA computer system*
Software engineering	*SE methodology, tools, and testing*
Integration, test, and support	*Integration of subsystems, testing, and support.*
Infrastructure	
Common software	*Development and maintenance of ALMA Common Software*
Preparation	
Observation preparation	*Development of ALMA observation preparation software, including tools for use by astronomers for preparing proposals and observations*
Scheduling	*Development of ALMA scheduling software*
Observing	
Control software	*Development of array monitor and control software, including antenna control*
Executive software	*Development of executive subsystem that coordinates the computing system*
Correlator software	*Development of ALMA correlator software, including embedded software*
Telescope calibration	*Development of synchronous calibration software*
Post Observing	
Pipeline	*Development of pipeline facility, including pipeline system and pipeline heuristics*
Archiving	*Development of archive software and system, including hardware, APIs.*
Off-line data reduction	*Development of software for standard off-line processing of ALMA observations, including engines to be used in pipeline*

Scope:

- Software
  - Engineering
  - Infrastructure
  - applications
- Hardware

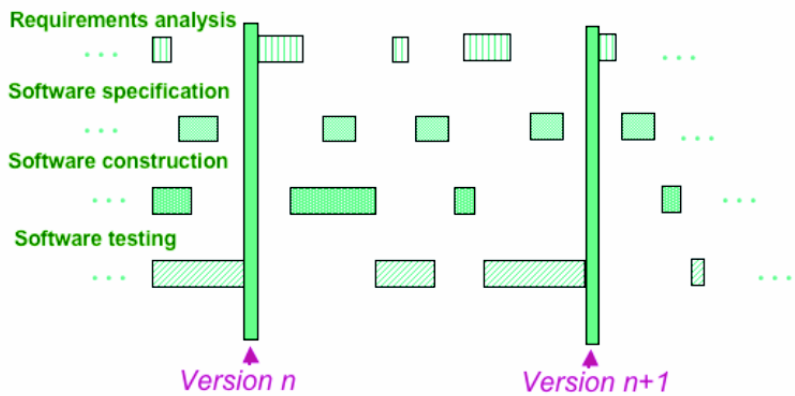
- Out scope:
- Calibration & imaging

Functionality and operational modes

- Requirements categories
  - Science goals and requirements
  - Engineering requirements (e.g. monitoring)
  - RFI mitigation
  - Data flow and data access patterns
  - Interoperability and virtual observatory style
- Operational model: open questions
  - Boundary between “instrument” and “user software”
  - Completeness of the software packages → function lists
  - Individual users or through science centers
  - ⇨ Inpact on software “flavour” and therefore on cost
  - ⇨ Close relation to operations TF

3.5. **Project requirements.** One of our Task Force members, when asked what could be done to ensure that the SKA software effort would be a success, replied: "Put all the developers in the same Square Kilometre".

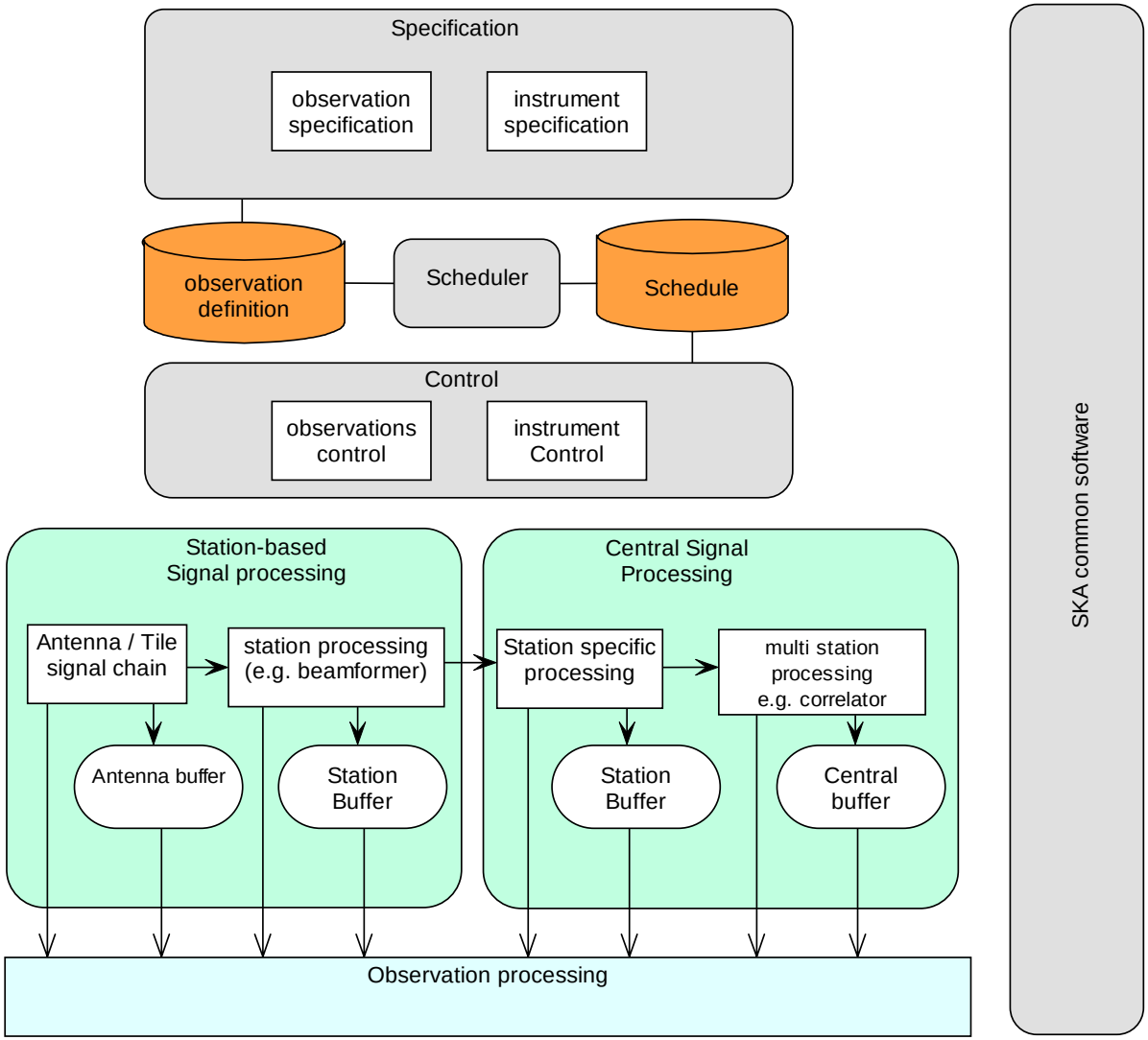
- Distributed development
- Architecture centric approach
  - Strong common software layers
  - Platform, low coupling, strong coherence etc.
- Development methodology
  - Adapt industry best practices
  - Probably evolutionairy (onion) or staged



- Distributed intelligence & autonomous operation
- Data flow and access patterns
- interoperability

- Cost drivers
  - Development cost driven by complexity and functional completeness
  - Hardware cost scales with FoV*BW*stations^2
  - Risk in distributed development
  - Very dependent on operation model issues
  - Some component costs are concept dependent

Analysis Architectural model



- Costs model:
  - Need in-depth analysis to bring to next level
  - there remains much to be defined - the scientific requirements, the physical concept, and the operational model are all unknown.
  - Need for operational model
- Concept selection:
  - System complexity drives cost
  - Esp. Control & monitoring and processing volume
- Scientific, engineering and other requirements:
  - Definition of the various requirements
  - proceed in tandem with the overall system design process.
- Industrial involvement:
  - We favor building partnerships with industry leaders

- Get grip on operational modes issues
  - Clearify where possible
  - Analysis impact in scenarios
- Gather requirements
- Next level in cost estimation
  - Bring architecture models one step further
    reference architecture
  - Per-component cost estimate; base on ALMA, LOFAR, KAT, xNTD, ATA,…..
  - Use industry connections to get estimates