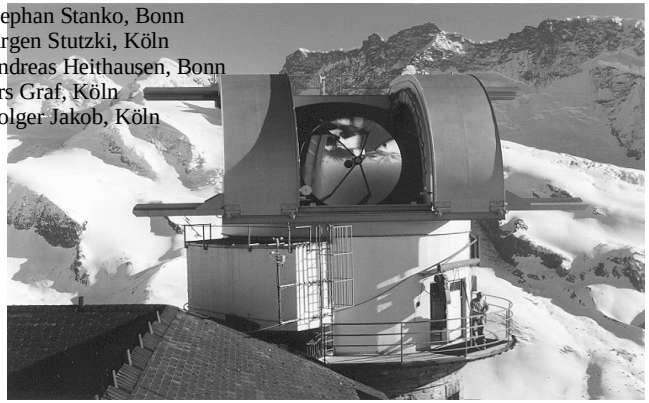
Kosma Control present status

Stephan Stanko, Bonn
Jürgen Stutzki, Köln
Andreas Heithausen, Bonn
Urs Graf, Köln
Holger Jakob, Köln



Outline

- Concept
- Software model
- Servers / software-hardware communication
- User interfaces
- Raw data and calibration
- Test facilities
- KOSMA_file_io
- Present status

Kosma Control design

- Intended as observers program for a telescope system
- Revision control system (CVS)
- Online documentation (cxref →HTML, LaTeX, RTF, ...)
- Uses unified parameter and communication interface, which has been in use at KOSMA for over 2 years (KOSMA_file_io)
- Debugging system with multiple levels implemented; simultaneous output to logfile
- Supports multiple backends
- Supports multiple frontends (with different frequency setups)
- Split up into several independent tasks, which can be run and tested individually
- Easy implementation of new hardware

Concept

- Software model
  - Servers running in idle mode until they are triggered
  - Only one task to communicate with the hardware
  - May be distributed on many PCs
  - Only standard compiler and libraries needed
  - Data is written „raw“
  - Calibration and display by stand-alone programs (CLASS, ...)
- User interface
  - Console mode
  - Scripts
  - GUI

Concept

- Computer hardware
  - Standard PCs with network interface
  - specific interface cards for hardware communication to frontend and backend
  - No additional hardware interface except TTL-signal for Signal-Reference phase/Status info for sky chopper
- Interprocess communication
  - KOSMA_file_io
  - Local file sharing / NFS on distributed systems

Measurement Modes

- Total power
- Position switch
- Beam switch
- Double beam switch
- Frequency switch
- On-The-Fly
- Temperature calibration (load)
- Frequency calibration (comb)
- ... ?

Servers

- aos_server
  - Communication with AOS hardware
  - Runs on PC which is directly connected to the backend hardware
  - Writes data directly to hard drive / RAID
  - Communicates to rest of system with KOSMA_file_io
- tp_server
  - Total power measurement
  - Can run on any PC, communicates only via KOSMA_file_io
  - Sets environment (Mirrors, Loads, Sky-Chopper, Comb, Zero)

Servers

- bs_server
  - Performs Beam Switch Measurement → Start sky-chopper
  - in other respects equal to tp_server
- fs_server
  - Frequency Switch Measurement → Tell reference synthesizer for LO to wobble
- comb_server
  - Make Frequency Calibration Measurement → Switches Comb generator on and off

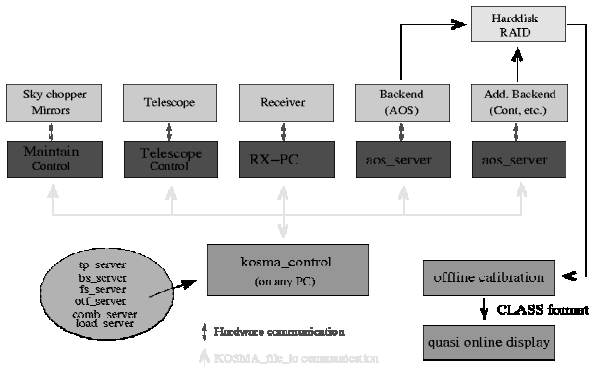
Servers

- load_server
  - Performs Temperature Calibration Measurement → Switches Loads
- otf_server
  - Differs from other measurement servers due to continuous data transfer
  - Commands aos_server **and** telescope to start measurement at a given time in the future (synchronization!)

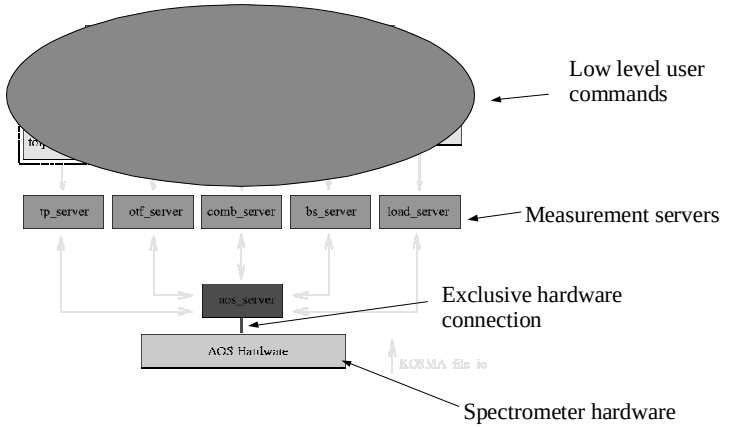
KOSMA's System
(as presently running on Gornergrat)

- Telescope and Sky-Chopper treated as „Black Boxes“ with KOSMA_file_io interface
- Array Receiver SMART with KOSMA_file_io interface
- Duas-SIS Receiver with DECnet communication, but accessible with KOSMA_file_io ← → DECnet server
- Array AOS with PC
- High Resolution Spectrometer (HRS), Variable Resolution Spectrometer (VRS), Medium Resolution Spectrometer (MRS) connected to one PC
- Continuum Backend (included into Array Receiver PC)

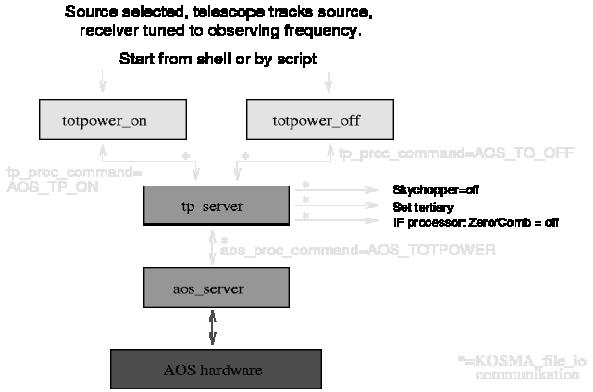
KOSMA's System – overview



Server model

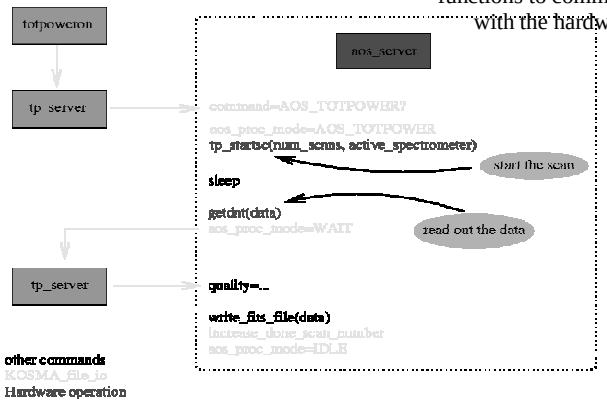


tp_server / aos_server



aos_server internals

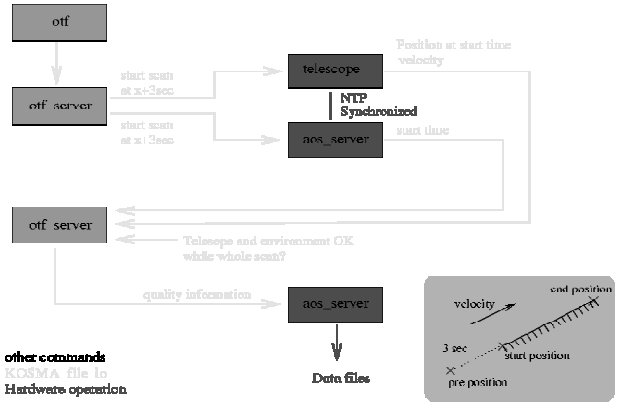
Only two well defined functions to communicate with the hardware



On-The-Fly mode

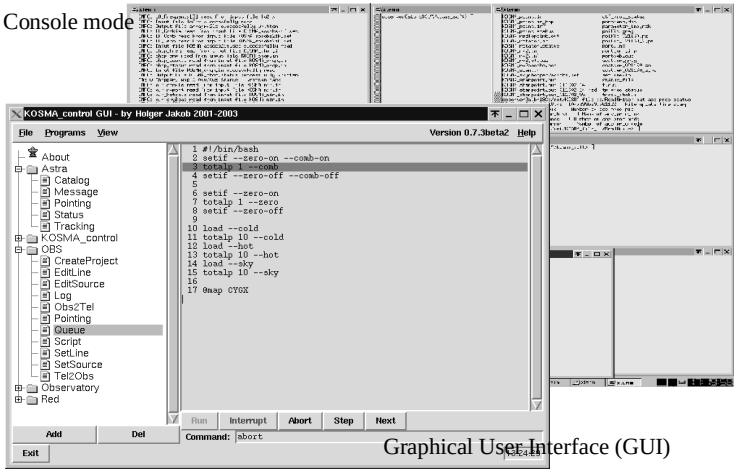
- PCs controlling backend and telescope have to be NTP synchronized
- aos_server (on backend) and telescope will be triggered a short time (1-3 seconds) in advance to be prepared
- Data (from backend) and position (from telescope) are marked with actual start-time for data/position correlation

On-The-Fly (OTF) mode



User interfaces

Console mode



Graphical User Interface (GUI)

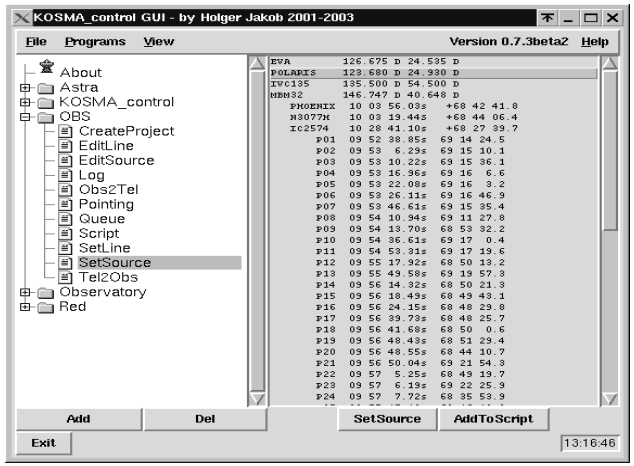
Console mode user interface

- Easy to implement
- Very flexible: scripting
- allows input from GUI
- System can be controlled remote, without graphic interface
- Starting and triggering of servers
- Changing of parameters
  - Editing KOSMA_file_io files with editor (e.g. edit observ_status, change entry int_time_on)
  - Specific variable manipulation programs (e.g. Kvar int_time_on=5)
  - additional parameters at measurement programs (e.g. totpoweron -int_time_on=5)
- Debugging by watching ASCII files

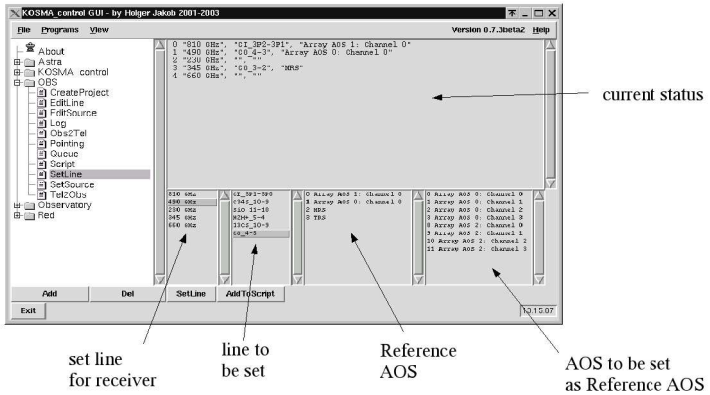
GUI (overview)

- Simple to use
- Status and warnings overview
- Useful for 'standard' measurement tasks (what is considered 'standard' will of course continuously expand)
- Uses console mode programs

GUI – Edit the source database

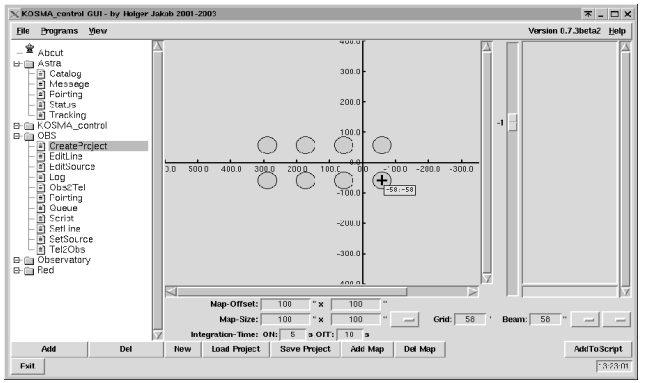


GUI – setting a line

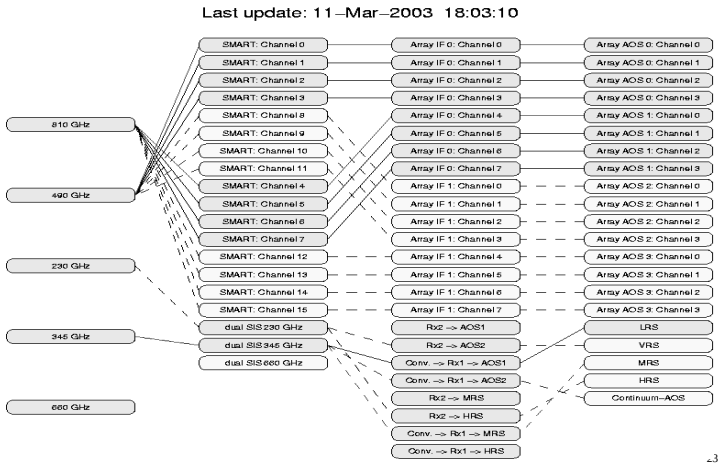


without GUI type: setline "490 GHz" CO_4-3

GUI – creating a project



Hardware status



Raw data and calibration

- Raw data is written into FITS files
- Each scan from each backend/channel is written into a separate file
- Measurement mode and scan numbers are written into the filename
- Example: 000254_CMB_Array_AOS01_Channel05.fits

Scan number
Measurement mode
Spectrometer name

- Simple FITS format
- Data is not calibrated
- 32 Bit data
- Header contains all information for data reduction

Raw data and calibration

(present KOSMA setup)

- Calibration and monitoring via stand alone offline calibration software
  - FITS to CLASS
  - works stand alone and quasi online
  - online display with CLASS
  - status monitor via KOSMA_file_io (tau, Trec, etc.)
- Measurement tasks (zero/hot/cold/sky/totalpower/otf/...) do not depend on each other

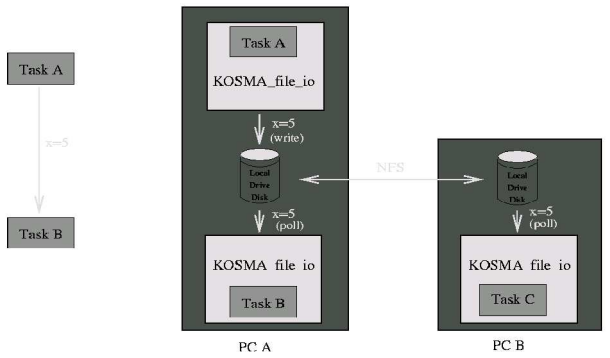
Test facilities

- Data acquisition only in aos_server
  → dummy data generation easily possible
- Offline debugging of all software-only problems
- Calibration (intensity and velocity) can be tested „offline“ (without actual hardware)
- Communication with all hardware components via KOSMA_file_io
  → dummy servers for sub-components like telescope, chopper, frontend etc.

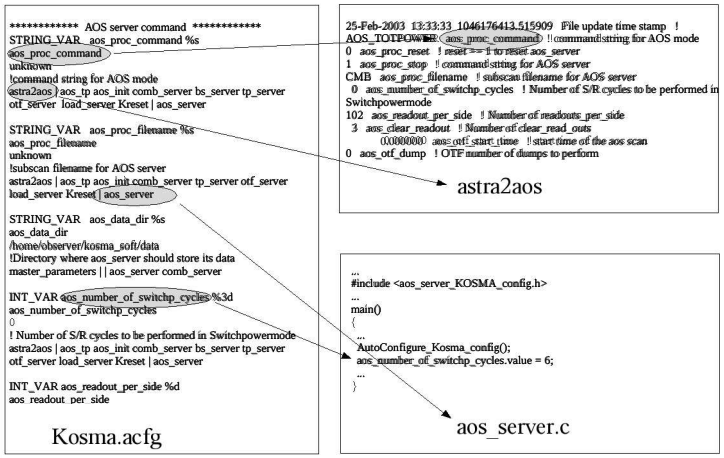
KOSMA_file_io

- Asynchronous communication via ASCII files
- makes debugging / monitoring very simple (editor, separate monitor program)
- Files are shared between PCs via standard NFS
- Synchronization possible through µs time stamps
- For faster transfer (reaction time less than 1s) a special file distribution system is available
- Configuration by one or more global script file(s) (KOSMA_config)
- Easy to use
- Easy to implement on different systems (Linux/Unix/NT)

KOSMA_file_io



Sample KOSMA_file_io files



Development status

- KOSMA_file_io successfully in use since over 2 years
- First tests of observing software at KOSMA successful in October 2002
  - aos-server ← → hardware
  - timing and controlling of the environment (telescope, ...)
  - measurment modes (total power, otf)
- All measurement modes are implemented
- Console mode needs some improvements
- Test mode in work/partly implemented
- Dummies available
- GUI in work
- Development under CVS since December 2002
- Documentation (cxref) partly implemented