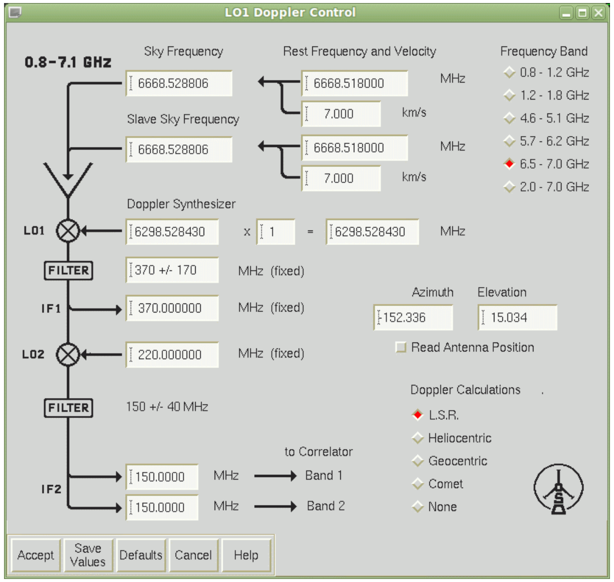
Observation Control Systems – the Onsala Perspective

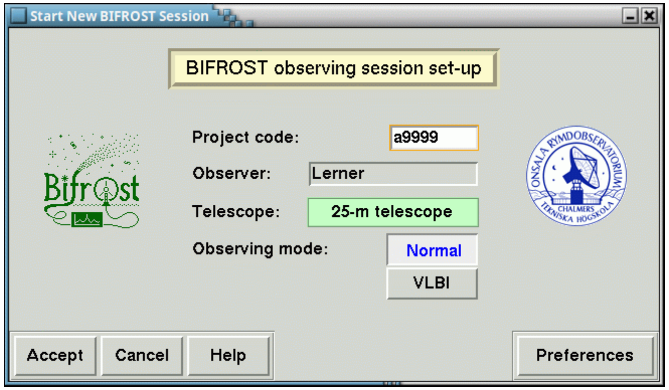
Mikael Lerner

Florence – 7 October 2015

Outline of presentation

- Personal background
- The Pegasus control system
- Thoughts on a new control system
- Towards the BIFROST control system



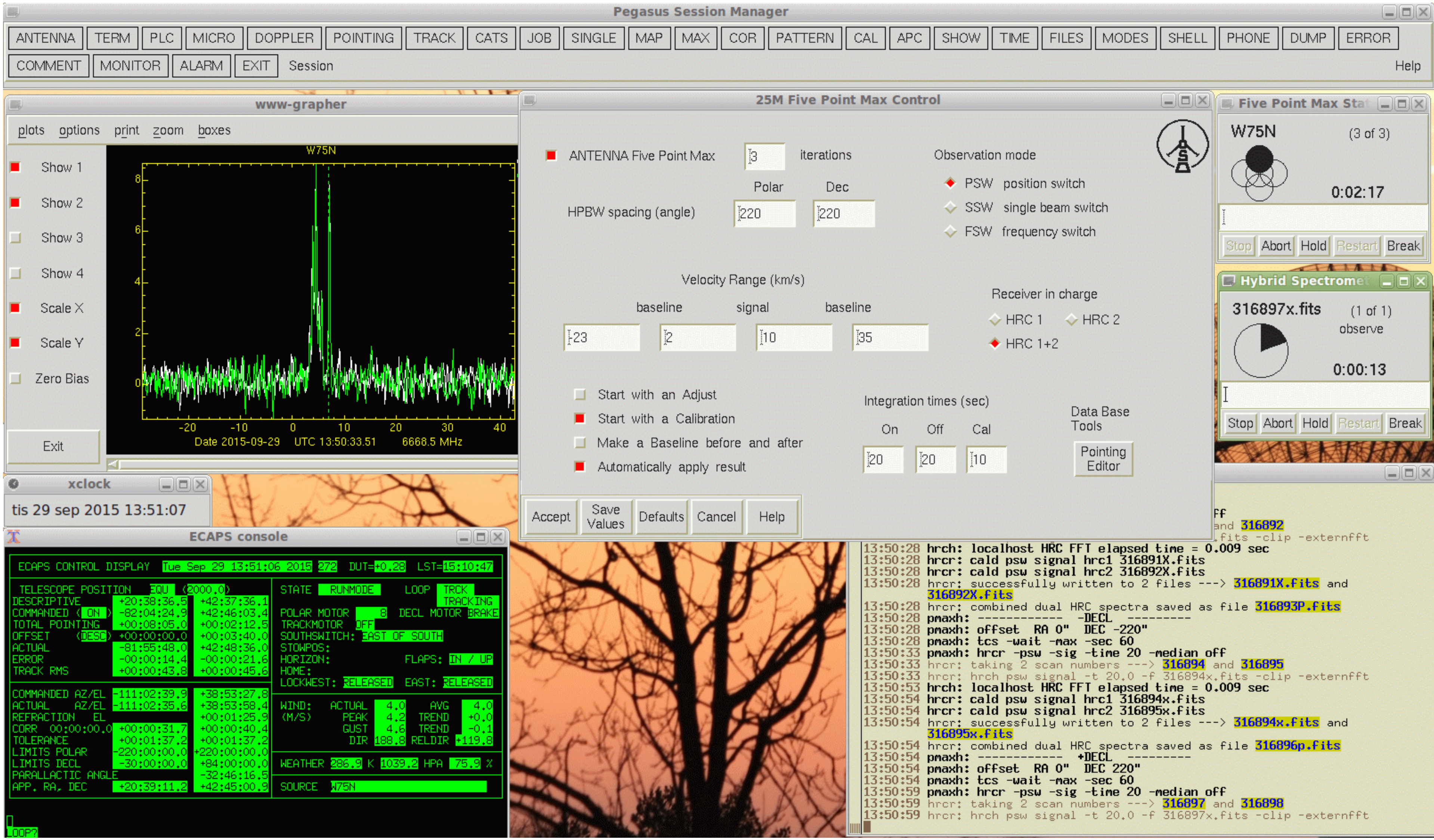


Personal background

- Received a Ph.D. in (numerical) astronomy at Onsala 1998
- Responsible for Pegasus control system at SEST 1999-2003
- Responsible for CIMA control system at Arecibo 2004-2009
- Responsible for Pegasus control system at Onsala 2009-present (two telescopes + ozone monitoring station)

Pegasus history

- Developed late 80s-early 90s at CFHT (but no longer used there)
- Imported to Onsala in early 90s and modified for radio telescopes
- Imported to SEST 1998 (used until closure 2003)
- Used at Onsala for 20m-telescope, 25m-telescope and ozone monitor station (the latter system replaced 2012)


Pegasus Session Manager
ANTENNA TERM PLC MICRO DOPPLER POINTING TRACK CATS JOB SINGLE MAP MAX COR PATTERN CAL APC SHOW TIME FILES MODES SHELL PHONE DUMP ERROR
COMMENT MONITOR ALARM EXIT Session
Help
www-grapher
plots options print zoom boxes
Show 1
Show 2
Show 3
Show 4
Scale X
Scale Y
Zero Bias
Exit
W75N
Date 2015-09-29 UTC 13:50:33.51 6668.5 MHz
25M Five Point Max Control
ANTENNA Five Point Max
3 iterations
Observation mode
PSW position switch
SSW single beam switch
FSW frequency switch
Polar Dec
HPBW spacing (angle) 220 220
Velocity Range (km/s)
baseline signal baseline
-23 2 10 35
Receiver in charge
HRC 1 HRC 2
HRC 1+2
Start with an Adjust
Start with a Calibration
Make a Baseline before and after
Automatically apply result
Integration times (sec)
On Off Cal
20 20 10
Data Base Tools
Pointing Editor
Accept
Save Values
Defaults
Cancel
Help
Five Point Max Sta
W75N (3 of 3)
0:02:17
Stop Abort Hold Restart Break
Hybrid Spectrome
316897x.fits (1 of 1)
observe
0:00:13
Stop Abort Hold Restart Break
xclock
tis 29 sep 2015 13:51:07
ECAPS console
ECAPS CONTROL DISPLAY Tue Sep 29 13:51:06 2015 272 DUT=+0.28 LST=15:10:47
13:50:28 hrch: localhost HRC FFT elapsed time = 0.009 sec
13:50:28 hrcr: cald psw signal hrc1 316891X.fits
13:50:28 hrcr: cald psw signal hrc2 316892X.fits
13:50:28 hrcr: successfully written to 2 files ---> 316891X.fits and 316892X.fits
13:50:28 hrcr: combined dual HRC spectra saved as file 316893P.fits
13:50:28 pmaxh: ------------ -DECL ---------
13:50:28 pmaxh: offset RA 0" DEC -220"
13:50:28 pmaxh: tcs -wait -max -sec 60
13:50:33 pmaxh: hrcr -psw -sig -time 20 -median off
13:50:33 hrcr: taking 2 scan numbers ---> 316894 and 316895
13:50:33 hrcr: hrch psw signal -t 20.0 -f 316894x.fits -clip -externfft
13:50:53 hrch: localhost HRC FFT elapsed time = 0.009 sec
13:50:54 hrcr: cald psw signal hrc1 316894x.fits
13:50:54 hrcr: cald psw signal hrc2 316895x.fits
13:50:54 hrcr: successfully written to 2 files ---> 316894x.fits and 316895x.fits
13:50:54 hrcr: combined dual HRC spectra saved as file 316896p.fits
13:50:54 pmaxh: ------------ +DECL ---------
13:50:54 pmaxh: offset RA 0" DEC 220"
13:50:54 pmaxh: tcs -wait -max -sec 60
13:50:59 pmaxh: hrcr -psw -sig -time 20 -median off
13:50:59 hrcr: taking 2 scan numbers ---> 316897 and 316898
13:50:59 hrcr: hrch psw signal -t 20.0 -f 316897x.fits -clip -externfft

Pegasus structure

- Mainly written in C using its own graphic library based on X
- A few framework programs are running continuously (main menu, feedback, communication daemons)
- Most programs are independent C-programs or shell scripts called from the main menu
- Information passing between programs is mainly done via variables in *par-files* (ASCII text files)
- Graphic layout is also defined in the same *par-files*
- Graphics is quite limited (B/W only, no lists or menus, no dynamic elements) forcing the use of other graphic solutions for more demanding windows (Tcl/Tk at SEST, Qt at Onsala)

Pegasus pros

- Fully menu-based GUI
- Simple, minimalistic graphic design
- Easy to learn and use (important since there are no telescope operators)
- The *Job*-facility provides means for simple scripting (for example pointing jobs)
- The map editor provides GUI allowing easy design of arbitrarily shaped maps
- Integrated quick-look display for the data

Pegasus cons

- Scripting possibilities are limited
- Too limited graphics capabilities within Pegasus framework
- Problems with coordination between different programs
- Bad error handling – bad variable values may block windows from appearing
- Difficult to install / no off-line demo-version available – requires modifications in operating system
- Complex structure with lots of “old luggage” that makes most modifications labour-intensive efforts

Wish list for a new control system

- Better scripting capabilities (sequential execution) – later also auto-observing mode (dynamic decision)
- Save configurations with automatic descriptions
- Reconfigurable for remote users (window size, update rate)
- Flexible options for feedback/logging with retroactive debug messages added upon errors
- Auto-sort and create meta-database of data taken
- More advanced quick-look facility with multiple plot windows and auto-configuration depending on data type
- Automatic syntax checking when loading user-generated objects (source lists, command scripts)
- All GUI features must have help texts (on-line manual)

Main design goal

BIFROST should provide everything that Pegasus provides and take user support to the next level.

Main design goals

BIFROST should provide everything that Pegasus provides and take user support to the next level.

I should be able to use the system at 04:00 AM.

Main design goals

BIFROST should provide everything that Pegasus provides and take user support to the next level.

I should be able to use the system at 04:00 AM. (without swearing)

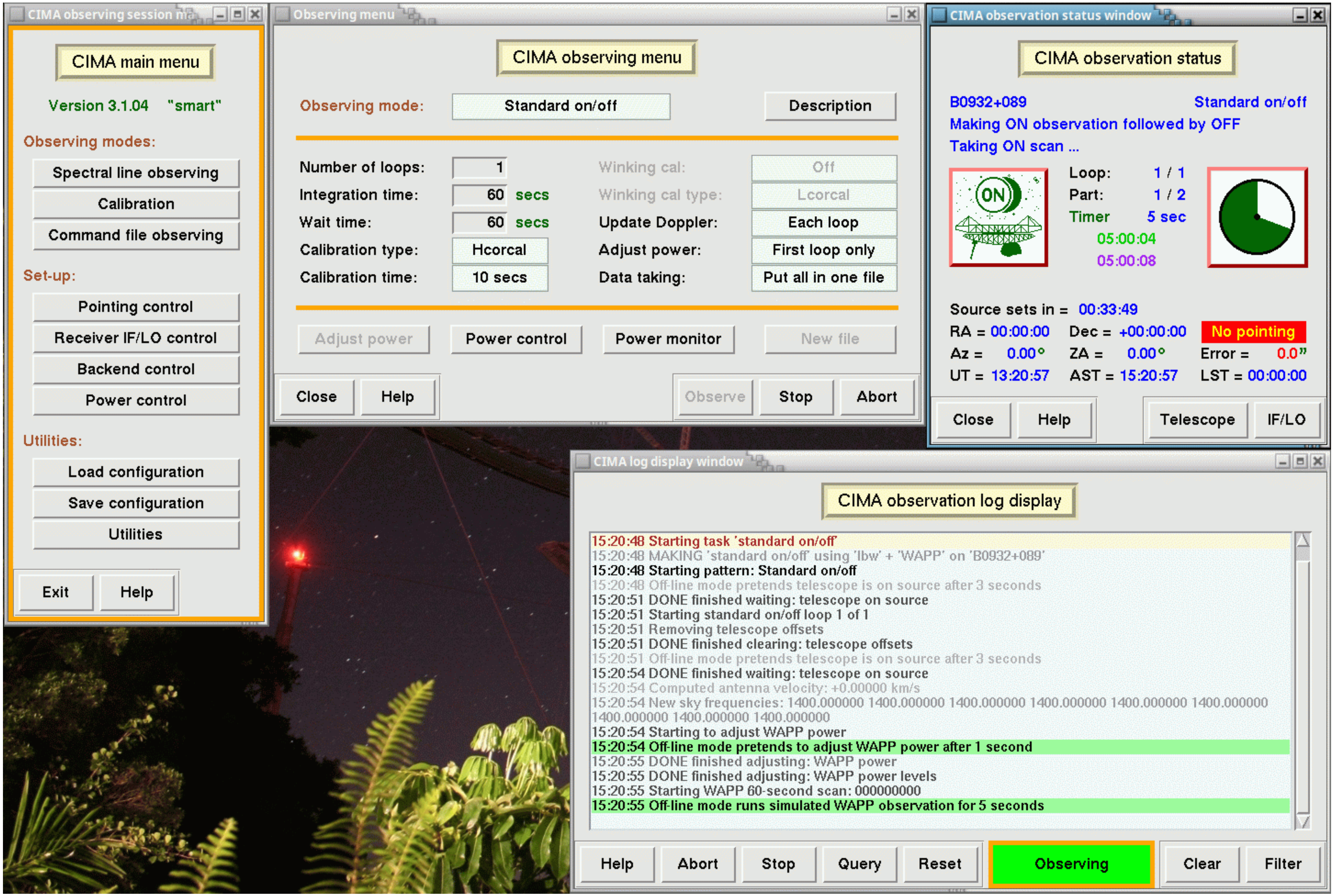
Design goals for BIFROST

- Intuitive, easy-to-use GUI needed
- On-the-fly input error checking
- Logical support (for example don't allow an observation to start if the backend has not yet been configured)
- Provide good awareness (what is the current status of the system)

- Use high-level scripting language
- Provide advanced graphics
- Use standard packages
- Avoid need for customization of operating system
- Be easy to maintain (even for non-programmers)

CIMA as a base for BIFROST

- Don't start from scratch: use CIMA imported from Arecibo as base for new control system
- Modern, flexible, script-based control system (Tcl/Tk)
- Tried and tested in challenging environment (many users, lots of different observing modes, remote users)
- Comes with a lot of desired features that Pegasus is lacking already included
- Large in-house knowledge about the system – no learning curve (major part of CIMA written by M. Lerner)


CIMA observing session
CIMA main menu
Version 3.1.04 "smart"
Observing modes:
Spectral line observing
Calibration
Command file observing
Set-up:
Pointing control
Receiver IF/LO control
Backend control
Power control
Utilities:
Load configuration
Save configuration
Utilities
Exit
Help
Observing menu
CIMA observing menu
Observing mode:
Standard on/off
Description
Number of loops: 1
Integration time: 60 secs
Wait time: 60 secs
Calibration type: Hcorcal
Calibration time: 10 secs
Winking cal: Off
Winking cal type: Lcorcal
Update Doppler: Each loop
Adjust power: First loop only
Data taking: Put all in one file
Adjust power
Power control
Power monitor
New file
Close
Help
Observe
Stop
Abort
CIMA observation status window
CIMA observation status
B0932+089
Standard on/off
Making ON observation followed by OFF
Taking ON scan ...
ON
Loop: 1 / 1
Part: 1 / 2
Timer 5 sec
05:00:04
05:00:08
Source sets in = 00:33:49
RA = 00:00:00
Dec = +00:00:00
No pointing
Az = 0.00°
ZA = 0.00°
Error = 0.0"
UT = 13:20:57
AST = 15:20:57
LST = 00:00:00
Close
Help
Telescope
IF/LO
CIMA log display window
CIMA observation log display
15:20:48 Starting task 'standard on/off'
15:20:48 MAKING 'standard on/off' using 'lbw' + 'WAPP' on 'B0932+089'
15:20:48 Starting pattern: Standard on/off
15:20:48 Off-line mode pretends telescope is on source after 3 seconds
15:20:51 DONE finished waiting: telescope on source
15:20:51 Starting standard on/off loop 1 of 1
15:20:51 Removing telescope offsets
15:20:51 DONE finished clearing: telescope offsets
15:20:51 Off-line mode pretends telescope is on source after 3 seconds
15:20:54 DONE finished waiting: telescope on source
15:20:54 Computed antenna velocity: +0.00000 km/s
15:20:54 New sky frequencies: 1400.000000 1400.000000 1400.000000 1400.000000 1400.000000 1400.000000 1400.000000 1400.000000 1400.000000
15:20:54 Starting to adjust WAPP power
15:20:54 Off-line mode pretends to adjust WAPP power after 1 second
15:20:55 DONE finished adjusting: WAPP power
15:20:55 DONE finished adjusting: WAPP power levels
15:20:55 Starting WAPP 60-second scan: 000000000
15:20:55 Off-line mode runs simulated WAPP observation for 5 seconds
Help
Abort
Stop
Query
Reset
Observing
Clear
Filter

Some goodies CIMA offers

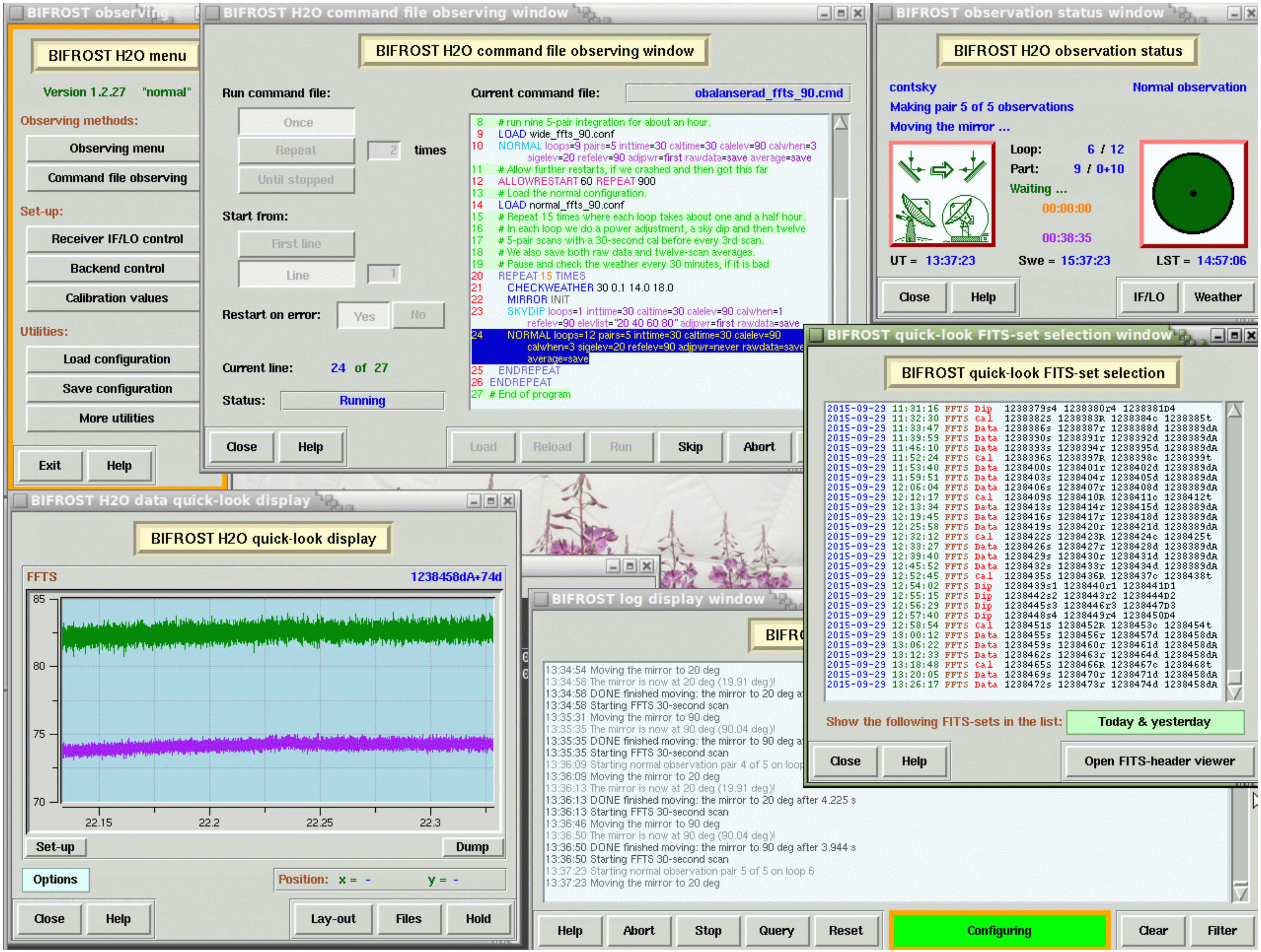
- Selection of font size (=window size) which is useful for remote observing
- Better command script facility
- Library functions for on-the-fly error checking of inputs
- Multiple versions available in parallel (e.g. new + stable)
- Off-line mode for training or preparation
- Flexible selection of what is logged and shown in the feedback as well as in what format
- *Shamecast*-support (*sha*red *me*mory multi*cast*) for getting instrument information
- Existing high-level procedures can be used as templates – thus adaptation instead of designing and writing from scratch

Towards BIFROST

- The BIFROST-platform has been created: a set of Tcl/Tk libraries with a standardized graphic design and special versions of *wish* and *tclsh* (standard Tcl/Tk linked with some extra libraries supporting *shamecasts*, astronomical calculations and Jeff Hagen's *socklib* package)
- Various peripheral systems are built on the BIFROST-platform: instrument *shamecasts*, monitor and web displays, engineering parameter logging, log viewers, alarms
- Gradual transition: several BIFROST-programs already used in Pegasus (spectral line selection, receiver tuning), other systems are BIFROST-compatible (VLBI-daemon)

Current BIFROST status

- The BIFROST-platform is used in a number of engineering systems: instrument monitoring, display windows, alarms
- A Pegasus system used for the ozone station was replaced by two BIFROST systems in 2012 (for parallel 24/7 operation of two independent instruments)
- A BIFROST system for the 25m telescope is ready for commissioning
- A BIFROST system for the 20m telescope could be available in 2016 (depending on priorities)


BIFROST H2O menu
Version 1.2.27 "normal"
Observing methods:
Observing menu
Command file observing
Set-up:
Receiver IF/LO control
Backend control
Calibration values
Utilities:
Load configuration
Save configuration
More utilities
Exit
Help
BIFROST H2O command file observing window
Run command file:
Once
Repeat
times
Until stopped
Start from:
First line
Line
Restart on error:
Yes
No
Current line: 24 of 27
Status: Running
Current command file: obalanserad_ffts_90.cmd
run nine 5-pair integration for about an hour.
LOAD wide_ffts_90.conf
Allow further restarts, if we crashed and then got this far
ALLOWRESTART 60 REPEAT 900
Load the normal configuration.
LOAD normal_ffts_90.conf
Repeat 15 times where each loop takes about one and a half hour.
In each loop we do a power adjustment, a sky dip and then twelve
5-pair scans with a 30-second cal before every 3rd scan.
We also save both raw data and twelve-scan averages.
Pause and check the weather every 30 minutes, if it is bad
REPEAT 15 TIMES
CHECKWEATHER 30 0.1 14.0 18.0
MIRROR INIT
ENDREPEAT
ENDREPEAT
End of program
Close
Help
Load
Reload
Run
Skip
Abort
BIFROST H2O observation status
contsky
Normal observation
Making pair 5 of 5 observations
Moving the mirror ...
Loop: 6 / 12
Part: 9 / 0+10
Waiting ...
00:00:00
00:38:35
UT = 13:37:23
Swe = 15:37:23
LST = 14:57:06
Close
Help
IF/LO
Weather
BIFROST quick-look FITS-set selection
Show the following FITS-sets in the list:
Today & yesterday
Close
Help
Open FITS-header viewer
BIFROST H2O quick-look display
FFTS
1238458dA+74d
Set-up
Dump
Options
Position: x = - y = -
Close
Help
Lay-out
Files
Hold
BIFROST log display window
13:34:54 Moving the mirror to 20 deg
13:34:58 Starting FFTS 30-second scan
13:35:31 Moving the mirror to 90 deg
13:35:35 Starting FFTS 30-second scan
13:36:09 Moving the mirror to 20 deg
13:36:13 DONE finished moving: the mirror to 20 deg after 4.225 s
13:36:13 Starting FFTS 30-second scan
13:36:46 Moving the mirror to 90 deg
13:36:50 The mirror is now at 90 deg (90.04 deg)!
13:36:50 DONE finished moving: the mirror to 90 deg after 3.944 s
13:36:50 Starting FFTS 30-second scan
13:37:23 Starting normal observation pair 5 of 5 on loop 6
13:37:23 Moving the mirror to 20 deg
Help
Abort
Stop
Query
Reset
Configuring
Clear
Filter

Conclusion

- Pegasus has been used at Onsala since early 90s
- Pegasus was innovative for its epoch, but it has limitations and it is now time to move on to the next level
- The CIMA system used at Arecibo has been selected as a base for the new BIFROST system
- BIFROST is expected to become the next-generation control system providing enhanced functionality in many aspects