# Automation Workshop

<u>Objectives of Workshop</u>

- Define meaning of automation
- Define requirements of LLRF and framework for implementation
- Define operator interfaces
- Define what needs to be automated
- Learn from existing experimental versions
- Identify what is missing and what we don’t know
- Develop possible architecture(s) for automation
- Develop strategic plan for automation development
- Agree on common framework for implementation of automation

# 1st DESY LLRF-Automation Workshop

Time and Date:
Monday, May 22th: 9:30-21:00
Tuesday, May 23th: 9:00-18:00
Wednesday, May 24th: 9:00-12:00

Location:
DESY, 3/304 (Helgoland)

Videoconference:
Available for WUT, TUL, SNS

Objectives of Workshop
- Define meaning of automation
- Define requirements for LLRF and framework for implementation
- Define what needs to be automated
- Learn from existing experimental versions
- Identify what is missing and what we don't know
- Develop possible architecture(s) for automation including interfaces to other subsystems and operator interfaces
- Develop strategic plan for automation development
- Agree on common framework for implementation of automation

Agenda:

*1. Session: Opening and Status (Chair A. Brandt)*

| | | |
|---|---|---|
| Mon | 09:30-09:45 | "Opening, Objectives: What should be automated?" (S. Simrock) |
| | 09:45-10:15 | "Ideas Concerning Automation" (M. Hoffmann) |
| | 10:15-10:45 | "Requirements for Automation (Strawman)" (S. Simrock) |
| | 10:45-11:00 | *Coffee Break* |
| | 11:00-12:00 | Discussion Automation |
| | 12:00-14:00 | *Lunch* |
| | 14:00-15:15 | "Status of Klystron FSM and Test-Results" (B. Koseda) |
| | 15:15-15:45 | "Status of LLRF FSM and Test-Results" (A. Brandt) |
| | 15:45-16:15 | "Overview Hera Automation" (S. Herb) |
| | 16:15-16:30 | Coffee Break |
| | 16:30-17:00 | "Overview SNS Automation" (K.-U. Kasemir, via VC) |
| | 17:00-19:00 | Discussion |
| | 19:00-21:00 | *Social Event* |

*2. Session: Objectives, Goal (Chair E. Vogel)*

| | | |
|---|---|---|
| Tue | 09:00-09:30 | "DOOCS and Automation" (K. Rehlich) |
| | 09:30-10:00 | "Improving Automation in a User Facility" (M. Huening) |
| | 10:00-10:30 | "Procedure-List" (S. Simrock) |
| | 10:30-11:00 | "Experience with TTF I Autom. and FLASH Operation" (V. Ayvazyan) |
| | 11:00-11:15 | *Coffee Break* |
| | 11:45-12:30 | Discussion (Architecture Development) |
| | 12:30-13:30 | *Lunch* |

*3. Session: Concepts, Ideas (Chair M. Hoffmann)*

| | | |
|---|---|---|
| Tue | 13:30-14:00 | "Automation in an Evolving Environments" (A. Brandt) |
| | 14:00-14:30 | "AI support for automation and automation design" (B. Koseda) |
| | 14:30-14:45 | *Coffee Break* |
| | 14:45-15:15 | "Database for Automation" (M. Wojtowski via VC) |
| | 14:45-17:00 | Discussion |

*4. Session: Outlook, Plan (Chair S. Simrock)*

| | | |
|---|---|---|
| Wed | 9:00-9:30 | "Summary of the Workshop" (A. Brandt) |
| | 9:30-10:00 | "Proposal for an Organization of the Work" (S. Simrock) |
| | 10:00-10:30 | Discussion |
| | 12:00 | End of the Workshop |

# Need for automation

**From ILC LLRF BCD :**

**10.7.5 Automation**

*The operation of the more than 560 linac rf systems will be highly automated by the implementation of a finite state machine finite state which has access to high level applications including the adjustment of the loop phase, vector-sum calibration, frequency and waveguide tuner control, and exception handling. The area of automation is viewed as a major area of R&D needed for the successful operation of the accelerator complex. The needs of RF system automation will help to define the structure and complexity of the control system.*

# Thesaurus

**Main Entry:** Automation

**Part of Speech:** *noun*

**Definition:** mechanization

**Synonyms:** computerization, industrialization, mechanization

**Source:** *Roget's New Millennium™ Thesaurus, First Edition (v 1.2.1)*
*Copyright © 2006 by Lexico Publishing Group, LLC. All rights reserved.*

# Other Definitions

- The art of making processes or machines self-acting or self-moving. Also pertains to the technique of making a device, machine, process or procedure more fully automatic.
- Making a process automatic eliminating the need for human intervention
- Substitution of human labour and skill with machinery, self-regulating devices or computers
- Automation (ancient Greek: = self dictated) or industrial automation is the use of computers to control industrial machinery and processes, replacing human operators. It is a step beyond mechanization, where human operators are provided with machinery to help them in their jobs. The most visible part of automation can be said to be industrial robotics. ...

- [,C:tE`meiFEn] (-ion 名词后缀)n.自动仪，自动控制

# RF Systems for XFEL

- RF Gun
- Injector
- 3rd harmonic cavity
- Main Linac

# Scope of RF Control

| | |
|---|---|
| total number of klystrons / cavities | ~ 30/ 1,000 |
| | |
| <u>per rf station (klystron):</u> | |
| # cavities / 10 MW klystron | ~ 32 |
| # of precision vector receivers (probe, forward, reflected power) | ~ 100 |
| # piezo actuator drivers / motor tuners | ~ 32/32 |
| # waveguide tuner motor controllers | ~ 32 |
| # vector-modulators for klystron drive | 1 |
| | |
| Total # of meas. / control channels | 3,000 / 3,000 |

# RF Control Requirements

- Maintain Phase and Amplitude of the accelerating field within given tolerances to accelerate a charged particle beam
  - up to 0.01% for amplitude and 0.01 deg.for phase
- Minimimize Power needed for control
- RF system must be reproducible, reliable, operable, and well understood.
- Other performance goals
  - build-in diagnostics for calibration of gradient and phase, cavity detuning, etc.
  - provide exception handling capabilities
  - meet performance goals over wide range of operating parameters

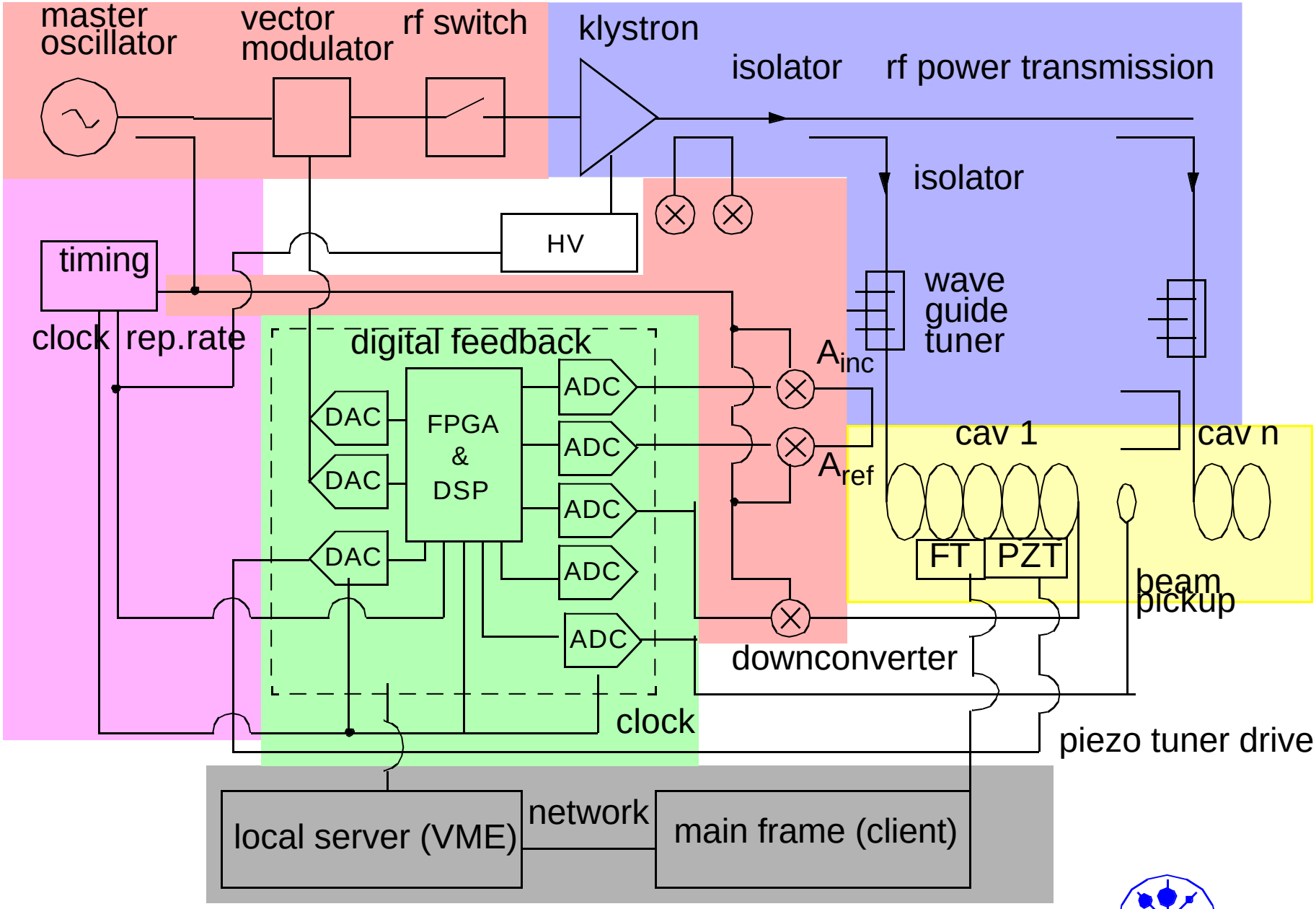
# Requirements RF Control

- Reliable
  - not more than 1 LLRF system failure / week
  - minimize LLRF induced accelerator downtime
  - Redundancy of LLRF subsystems
  - ...
- Operable
  - “One Button” operation (State Machine)
  - Momentum Management system
  - Automated calibration of vector-sum
  - ...
- Reproducible
  - Restore beam parameters after shutdown or interlock trip
  - Recover LLRF state after maintenance work
  - ...

# Requirements RF Control

- Maintainable
  - Remote diagnostics of subsystem failure
  - “Hot Swap” Capability
  - Accessible Hardware
  - ...
- Well Understood
  - Performance limitations of LLRF fully modelled
  - No unexpected “features”
  - ...
- Meet (technical) performance goals
  - Maintain accelerating fields - defined as vector-sum of 32 cavities - within given tolerances
  - Minimize peak power requirements
  - ...

# Architecture of digital RF Control



**A Frequency generation**
(1) Stable reference frequency oscillator
(2) Phase locked Oscillator (various frequencies)
(3) Power supply
(4) Diagnostics
(5) Control system interface

**B. Frequency and Reference Phase Distribution**
(1) Phase stable transmission line
(2) Temperature stabilization
(3) Power distribution (directional couplers)
(4) Phase stability monitoring and correction

**C. Cavity Field Control**
(1) Detectors for accelerating field
(a) downconverter
(b) A&P detector
(c) I/Q detector
(2) Controllers for klystron drive
(a) A&P modulator
(b) vector-modulator
(3) Digital Feedback/Feedforward
(a) Fast analog IO (ADC/DAC)
(b) Signal Processors (FPGA,DSP)
(4) Feedback/Feedforward Algorithms
(5) Interlock system
(6) Diagnostics
(7) Interface to control system

**D. High Power Amplifier**
(1) RF power source
(2) Power supply
(3) Interlocks
(4) Diagnostics
(5) Interface to control system

**H. Machine Protection System**
**I. Personnel Safety System**
**J. Control System Interface**

**G. Cavity Frequency Tuning System**
(1) Cavity tuner (fast and/or slow)

**E. Power Transmission System**
(1) Transmission line (coaxial, waveguide)
(2) Circulator, Isolator
(3) Power dividers
(4) Directional coupler (Monitor)
(5) Waveguide (coaxial) window
(6) Pressurisation system

**F. Accelerating System**
(1) Cavity
(2) Fundamental Coupler
(3) Higher Order Mode Coupler



# Linac RF Subsystems

# LLRF Control Algorithms

## A. FIELD CONTROL ALGORITHMS

(1) Feedback
- (a) PID filter
- (b) Kalman filter
- (c) adaptive filters
- (d) optimal controller

(2) Feedforward
- (a) beam loading compensation

(3) Beam based feedbacks
- (a) rf phase feedback
- (b) beam energy feedback
- (c) bunch length feedback

(3) Klystron linearization

(4) Exception handling
- (a) quench detection and handling
- (b) error from beam loading

## B. LLRF System Measurement Algorithms

(1) Loop phase rotation matrix

(2) Field calibration rotation matrix
(based on rf, beam based transients, and spectrometer)
- (a) gradient calibration
- (b) phase calibration

(3) Vector-sum calculation

(4) Meas. of incident phase (vector-sum !)

(5) Beam phase measurement

(6) forward/reflected power calibration
- (a) correct for directivity of couplers

(7) Cavity detuning
- (a) average during pulse
- (b) detuning curve during pulse

(8) Loaded Q

# LLRF Control Algorithms

**D. High level procedures**

(1) Adaptive feedforward
- (a) response matrix or T.F. based
- (c) robustness
- (d) different beam modes

(1) System identification
- (a) beam phase and current
- (b) loaded Q
- (c) incident phase

(3) Waveguide tuner control

(4) Momentum management system

(5) Field control parameters optimization

(6) Operation at different gradients

(7) Operation at the performance limit
- (a) maximize availability
- (b) maximize field stability

(8) Hardware diagnostics

(9) On-line rf system modelling

(10) Automated fault recovery

(11) Finite state machine

**C. Cavity Resonance Control**

(1) Slow tuner
- (a) maintain average resonance frequency (pre-detuning)
- (b) maximize tuner lifetime

(2) Fast tuner (ex. piezoelectric tuner)
- (a) dynamic Lorentz force compensation
- (b) microphonics control
- (c) minimize rf power required for control

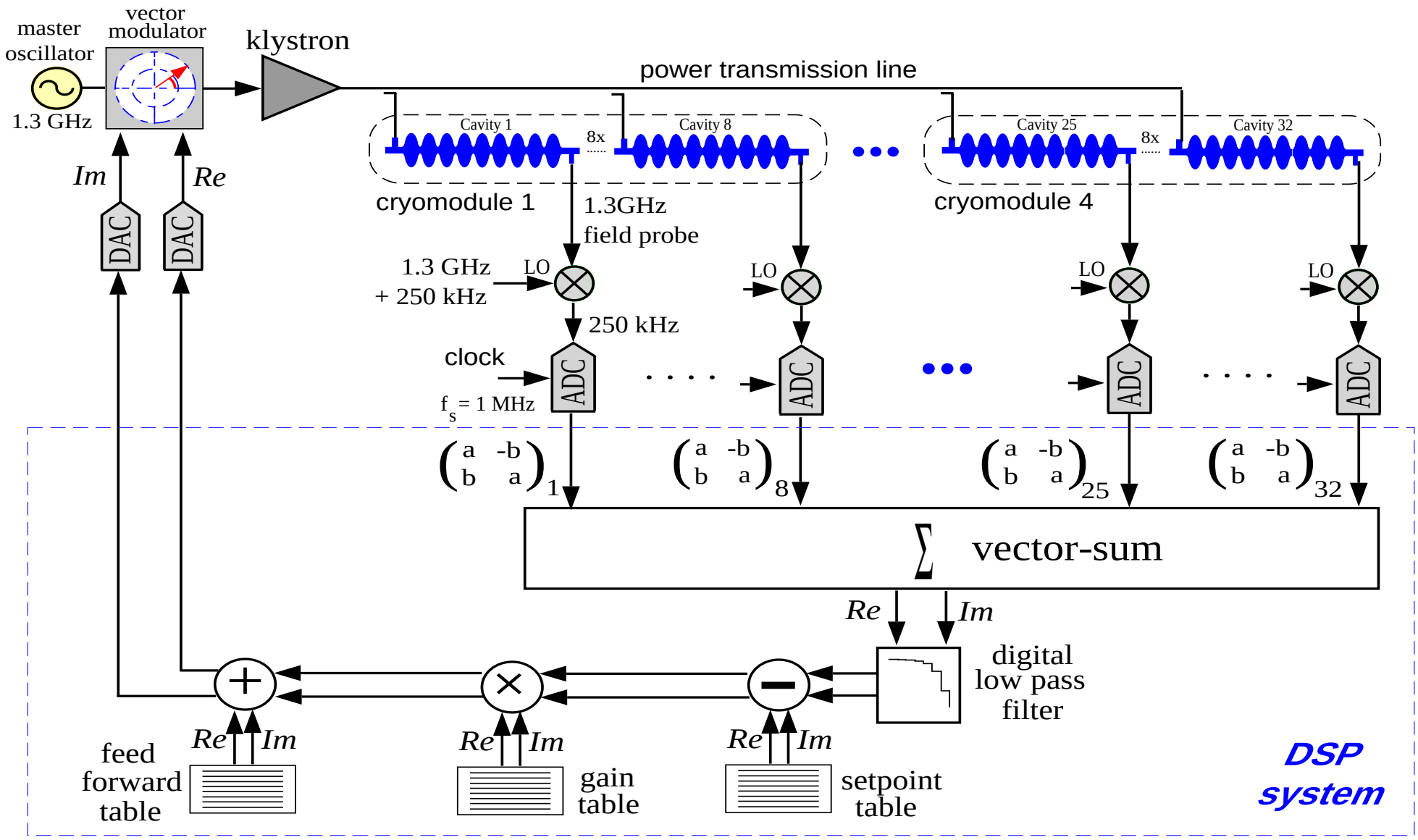
**E. Other**

(1) RF System Database
- (a) calibration coefficients
- (b) subsystem characteristics

(2) Alarm and warning generation

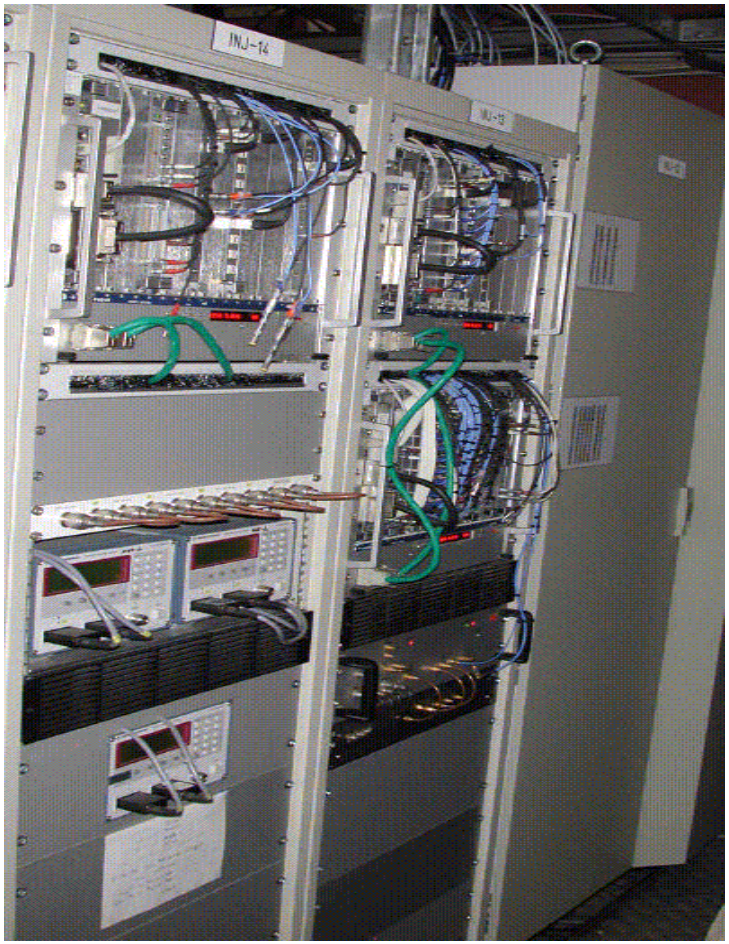
(3) Control System functions

# Digital Control at the TTF

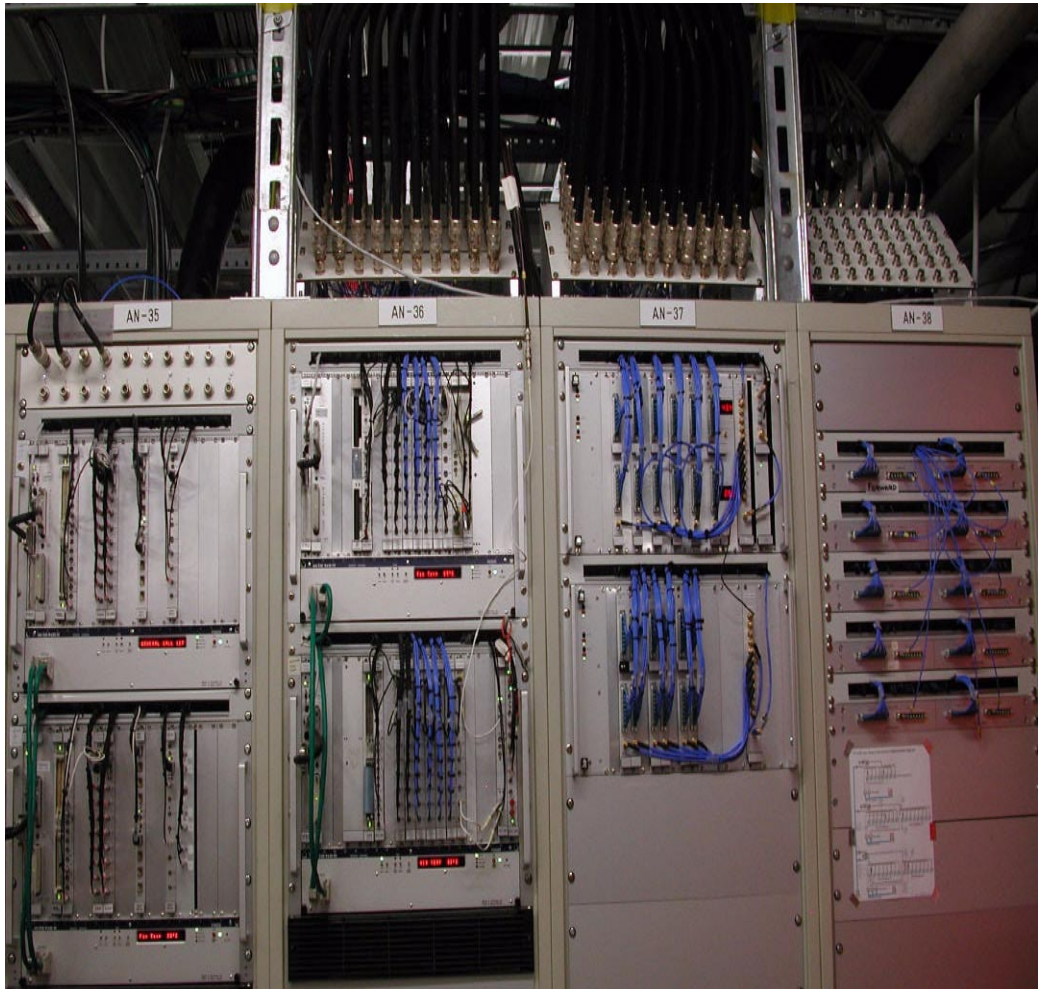


# Digital Feedback Hardware

Gun and ACC1




ACC2, ACC3, ACC4 & ACC5



# Choices of Automation

- Sequencer
- Event handler
- Finite State Machine
- Stateflow

# What needs to be automated

- Procedure (sequence of complex actions), example: turn on rf
- Application (complex action or calculation), example: adaptive feedforward
- Algorithm (calculation; signal_in => signal out); example: measure cavity detuning
- Function (simple action); example: turn on/off rf switch
- Signals and Parameters

# Elements of FSM

- Define
  - Superstates, simple states (hierachy, parallelism, history)
  - Flows
  - Conditions
  - Events
  - Signals and Parameters

# Types of Signals

**Signals**

- physical signals measured (analog and digital)
- physical signal calibrated (analog and digital)
- derived signals measured
- raw control signals
- derived control signals
- system parameters
- timing
- events
- warnings
- alarms
- llrf interlocks
- other subsystem interlocks

# Procedures

- rf commissioning
  - Initial (first time)
    - Hardware/software diagnostic
    - Level adjustment (probe forward, reflected)
    - Calibrate gradient/forward/reflected power
    - Cavity tuning (few hundred kHz)
    - Operation with cavity simulator
    - Downconverter calibration with test signals
    - RF reference check
  - after long shutdown
    - Hardware/software diagnostic
    - Measurement chain check
    - Operation with cavity simulator
    - Low power check
    - Verify calibrations and operational parameters
  - after maintenance day
    - Hardware/software diagnostic

# Procedures (Cnt'd)

- rf operation
  1. Turn on relevant subsystems and set their parameters
  2. set low power operation in open loop mode and verify signals
  3. Turn-on feedback and optimize feedback parameters
  4. Set-up for short beam pulses and operate with beam
  5. Set-up for long beam pulses
- rf system maintenance
  - Hardware/software diagnostic
  - Operation with cavity simulator and signal integrity check
  - Measure rf test signals and identify defect channels
  - Measure maximum operable gradients
- rf system debugging and trouble shooting
  - Measure rf test signals and identify defect channels

# Procedures (Cnt'd)

- special procedures
  - coupler warm and cold conditioning
  - cavity conditioning
- control of llrf related subsystems
  - master oscillator and frequency distribution
  - LO- oscillator generation
  - timing system (trigger, clocks)
  - klystron and modulator
  - Cryogenics
  - cavity, coupler and frequency tuner
  - machine protection

# Procedures (Cnt'd)

- special procedures
  .. coupler warm and cold conditioning
  .. cavity conditioning

- control of llrf related subsystems
  .. master oscillator and frequency distribution
  .. LO- oscillator generation
  .. timing system (trigger, clocks)
  .. klystron and modulator
  .. cryogenics
  .. cavity, coupler and frequency tuner
  .. machine protection

# Applications

- Autotune (motor tuner)
- By-passing of cavities
- Adjust power levels waves
- Verify LO/clock signal integrity
- Adaptive feedforward
- System Identification
- Momentum Management

# Algorithms

- Measure cavity detuning and load Q
- Measure loop phase and loop gain
- Calibrate gradient and phase, forward and reflected wave
- Measure klystron saturation curve
- Cavity field estimator
- RMS amplitude and phase measurement
- Calculate available gradient/current headroom

# Requirements for Automation

- Framework for implementation must
  - be modular (it must be easy to add procedures)
  - easy to maintain and to troubleshoot
  - easy to learn
  - automatically adapt to ‘state’ of accelerator
  - Support development of new algorithm by permitting direct access to all parameters
  - Allow operational changes
  - be compatible with XFEL Control System
  - well documented
  - Provide interfaces for other subsystems and operators (especially higher level FSM)

# Prerequisites for Automation

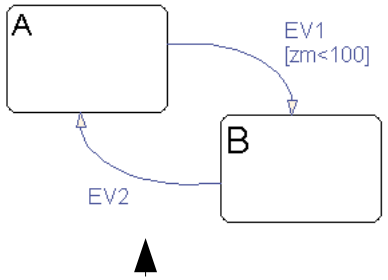
- Well understood LLRF system
- Documentation of all procedures and signals
- Framework for implementation defined and available
  - Communication protocols
  - Graphical and textinterface for implementation

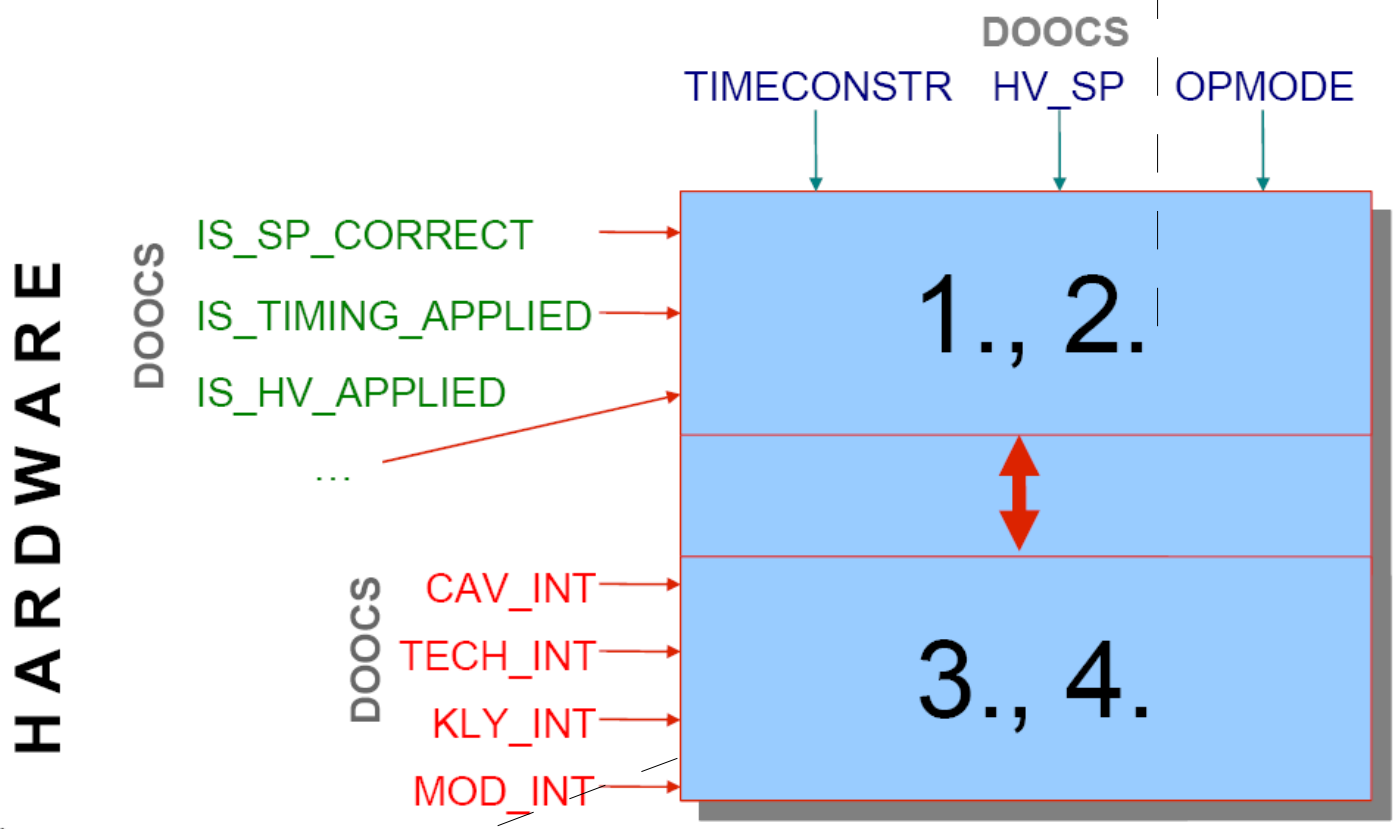
# Automation/decision support system for RF-power stations.

Bogusław Kosęda – DMCS/TUL

**Solution should be able to:**

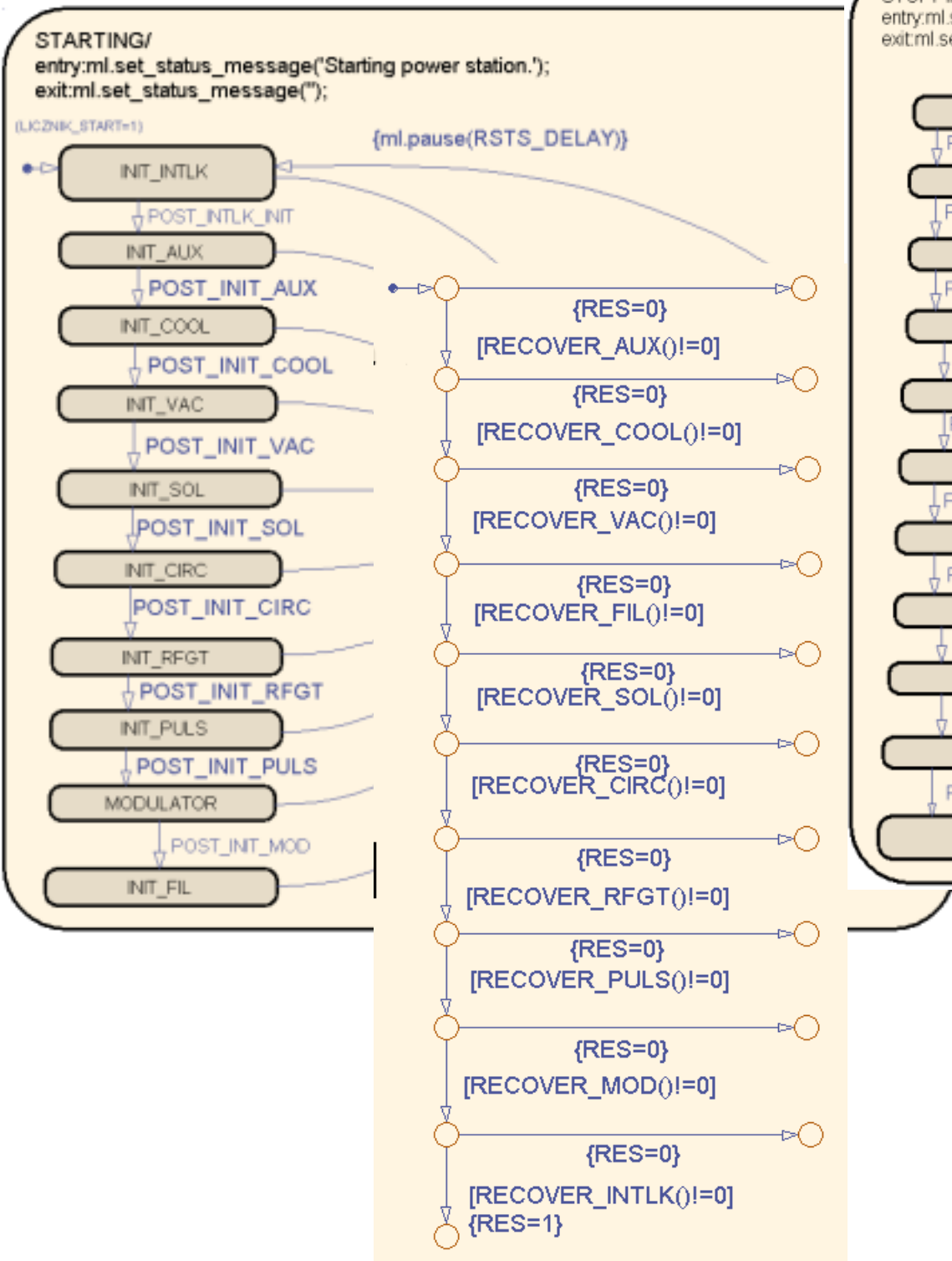
1. Invoke routine procedures on behalf of the operator start/stop/reset/restore setpoints.
2. Keep the machine in preconfigured regime of work.
3. Handle the exceptions.
4. Realize auto recovery procedure.

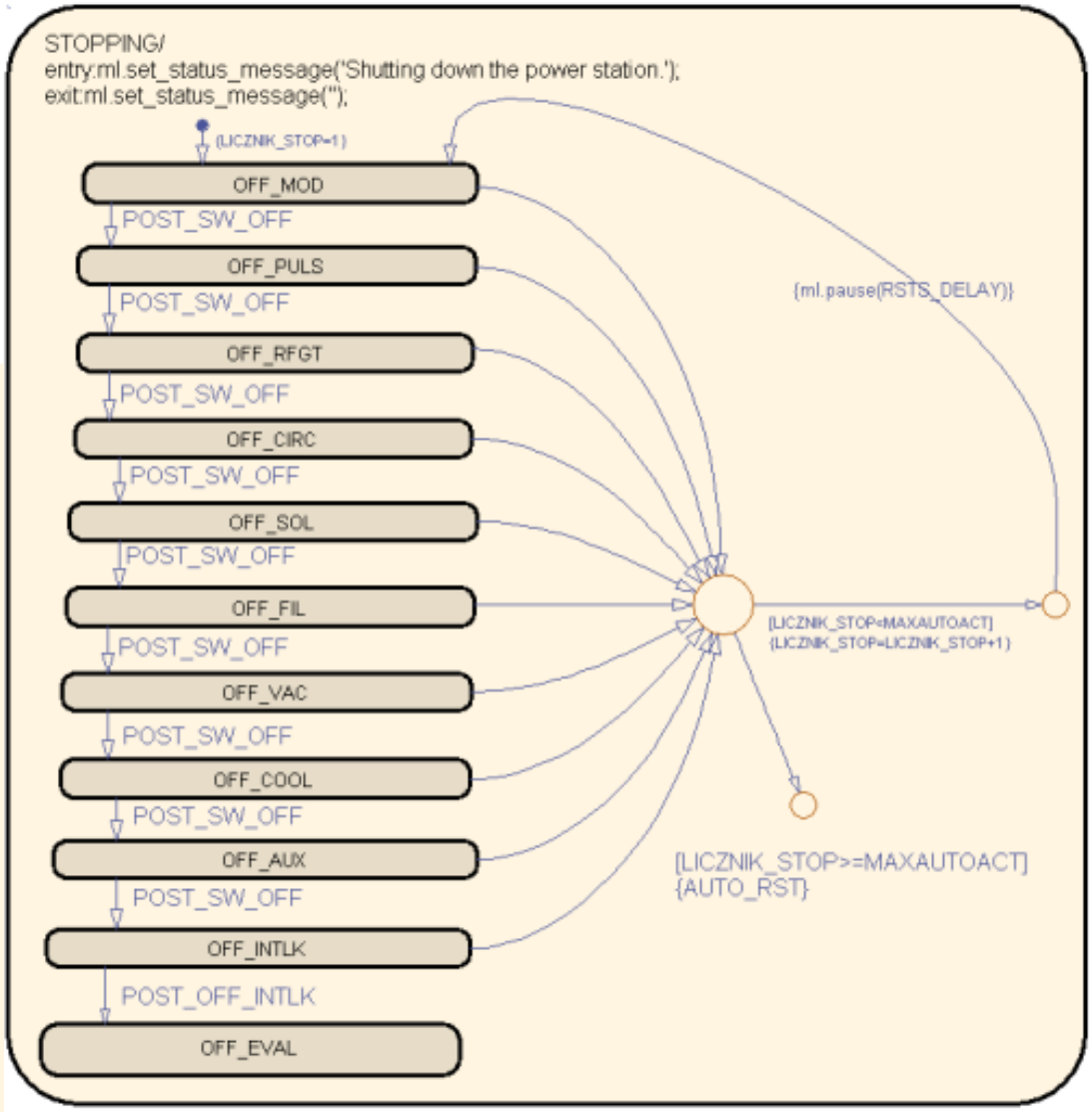


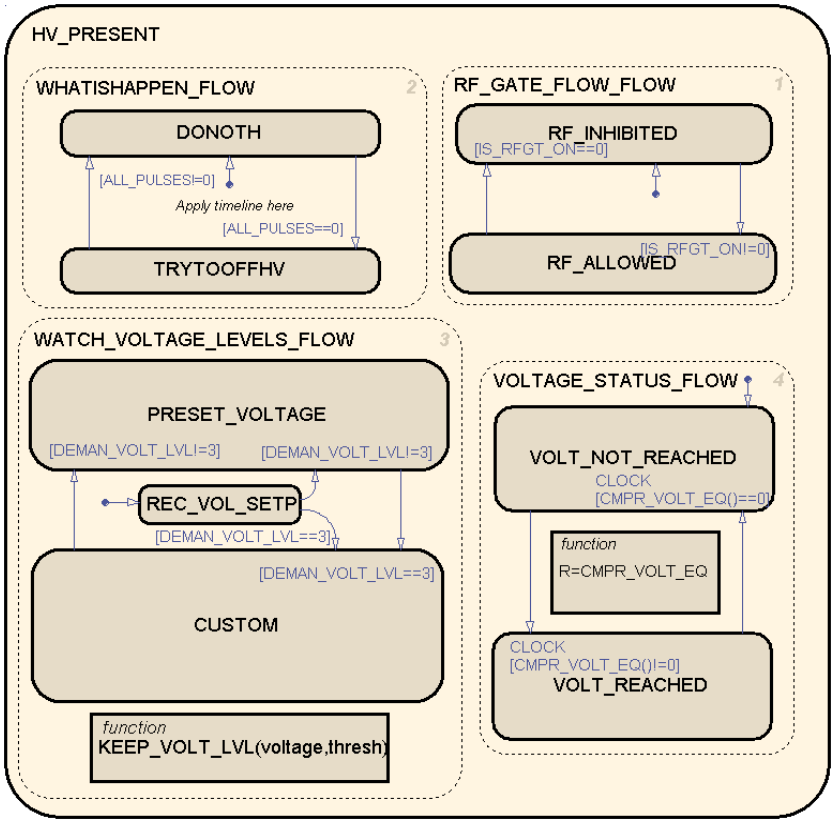

```
%Actions
signal(63,'RF-Inhibited',action).
signal(59,'Preamplifier (ready status)',action).
%Warnings
signal(16,'Ion pump current attention',warning).
signal(15,'Oil level of the klystron (attention)',warning).
%Compound interlock patterns
evt_pattern(104,action,'User defined pattern 3',[1,3],[beforeq(1,3)]).
```

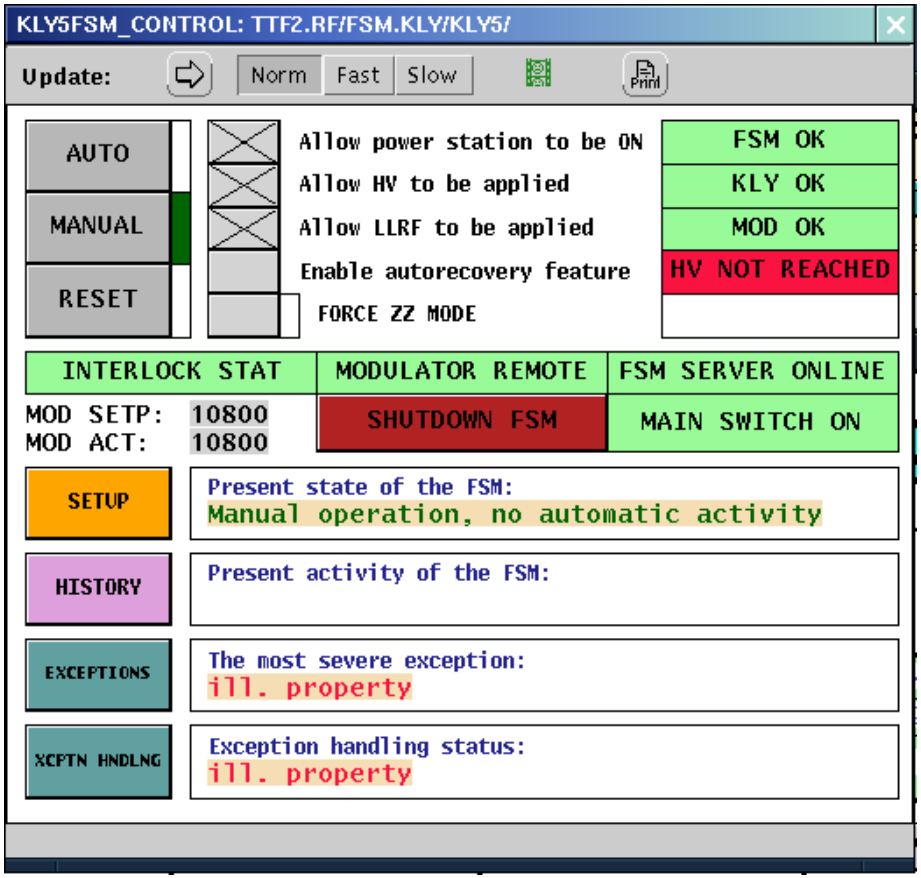
# Invoking procedures on behalf of the operator

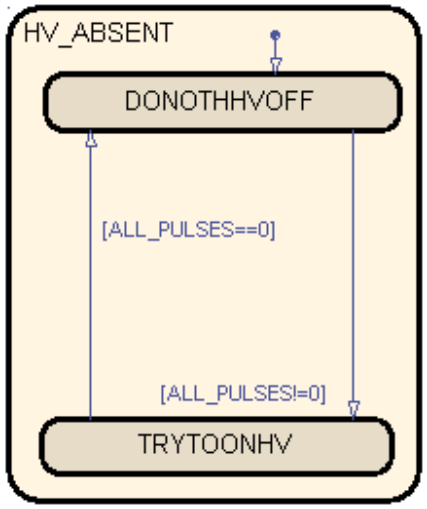


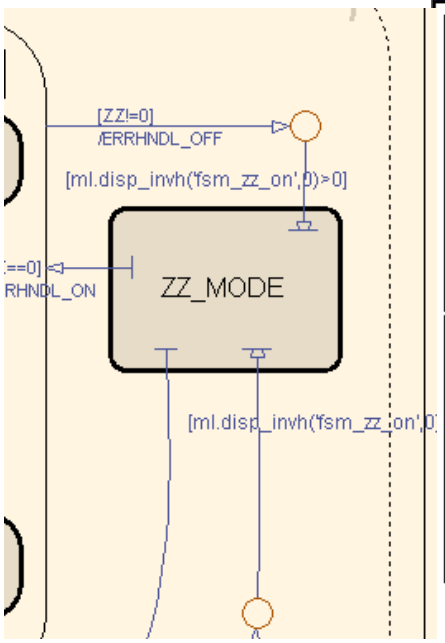



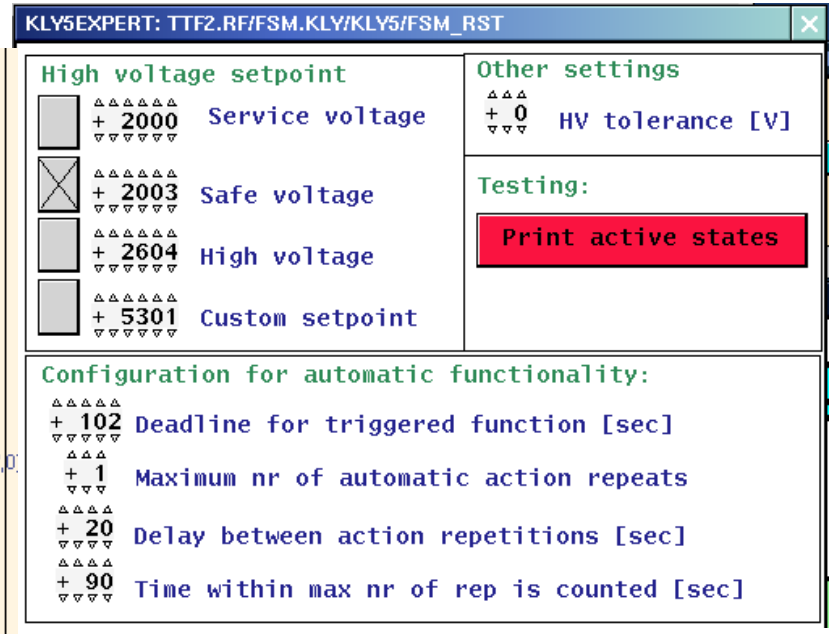
# Keeping the machine in preconfigured regime of work



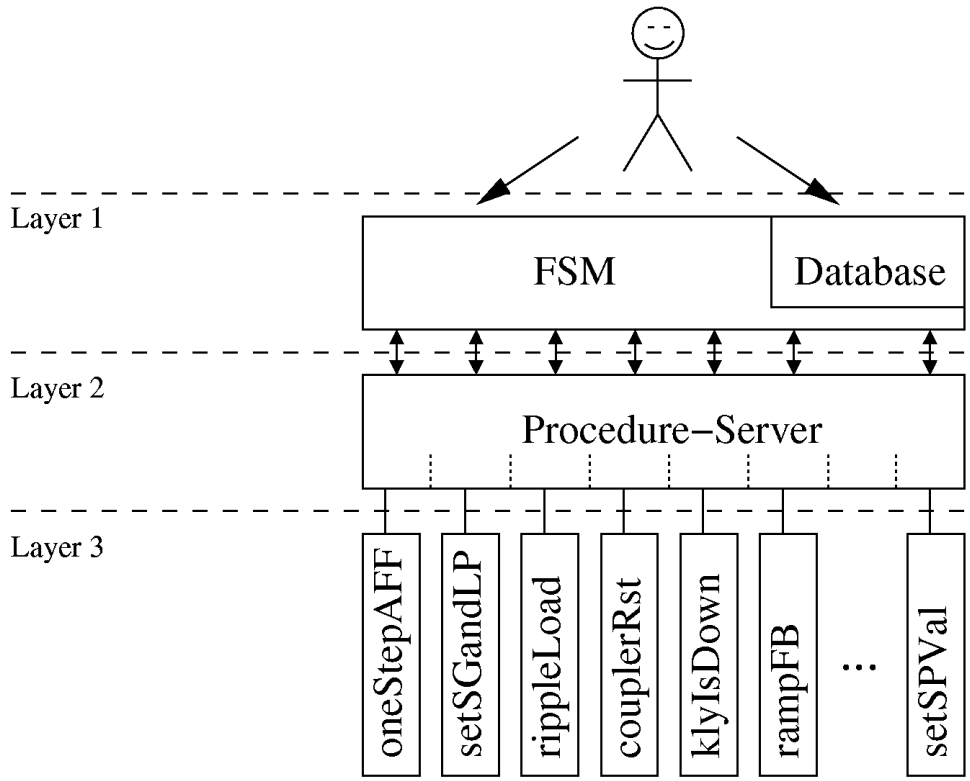



# Status of LLRF FSM and Test Results

alexander.brandt@desy.de

# FSM-Architecture




## Procedures

- Compiled scripts of any programming language (and any Matlab-version)
- "Fire and forget":
  - –Invoke - Run - Return
  - –"Stateless Procedures"
- E.g.:
  - –Adaptive Feedforward
  - –Loop-Phase
- Web-Documentation
- Algorithms are identical for all RF-stations and read a config file

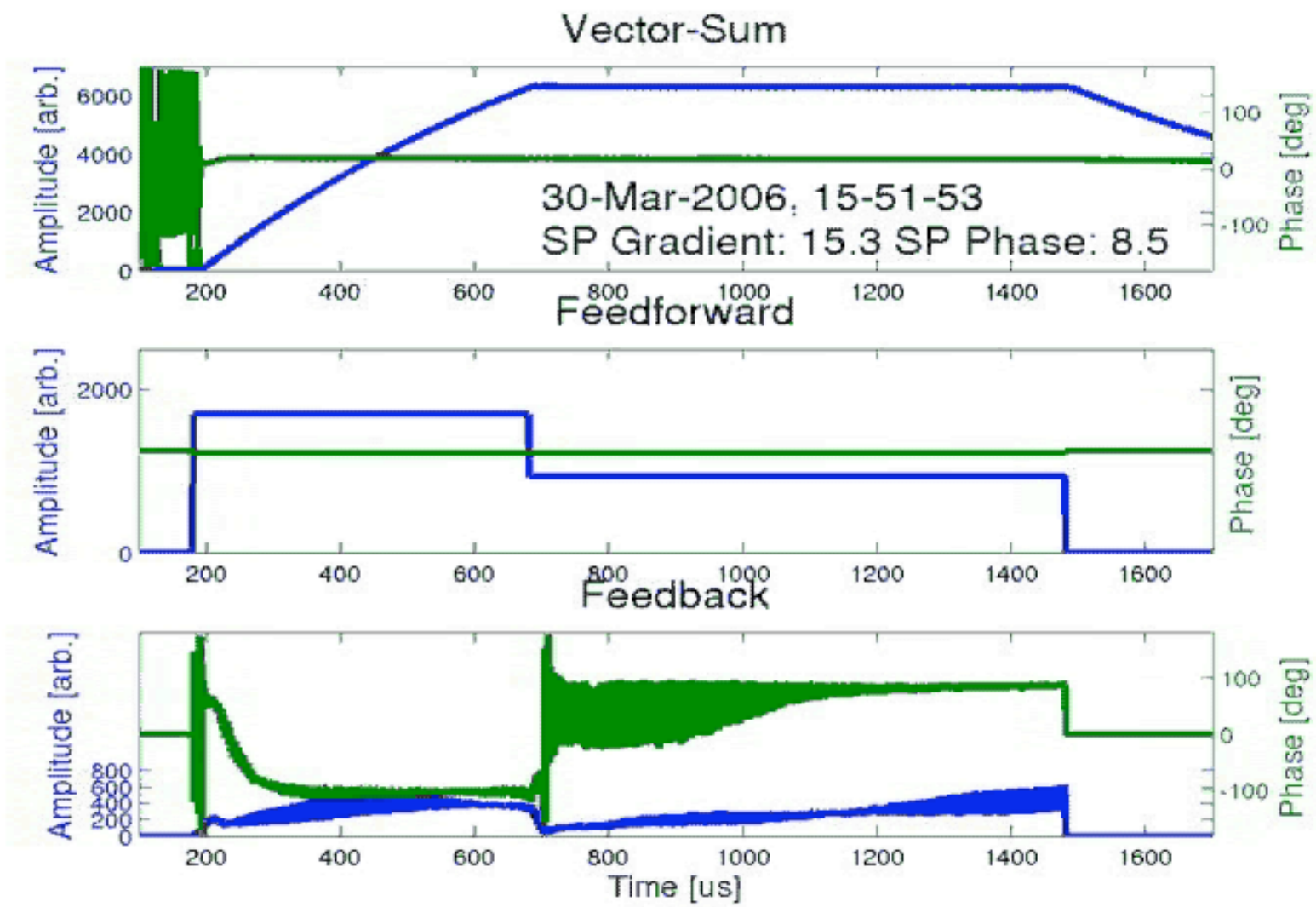
# Procedure Server

Hist 0-15
Hist 16-31

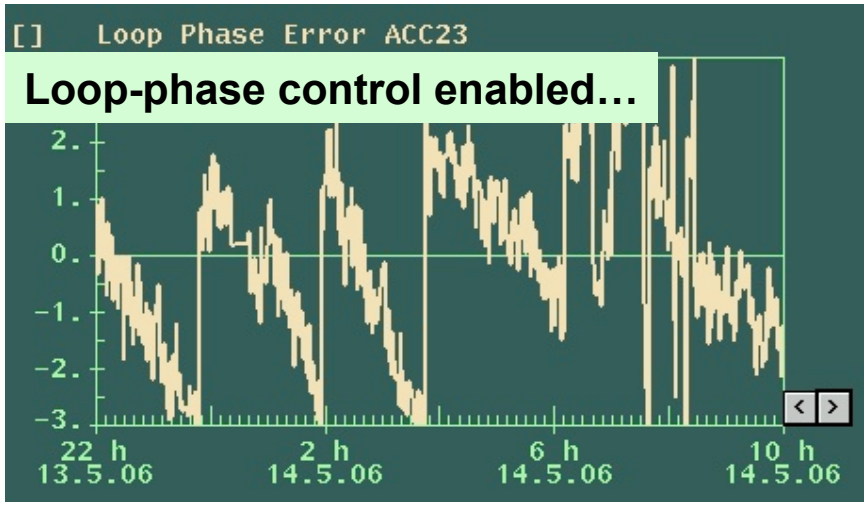
## ``Stateless Procedures''-Server

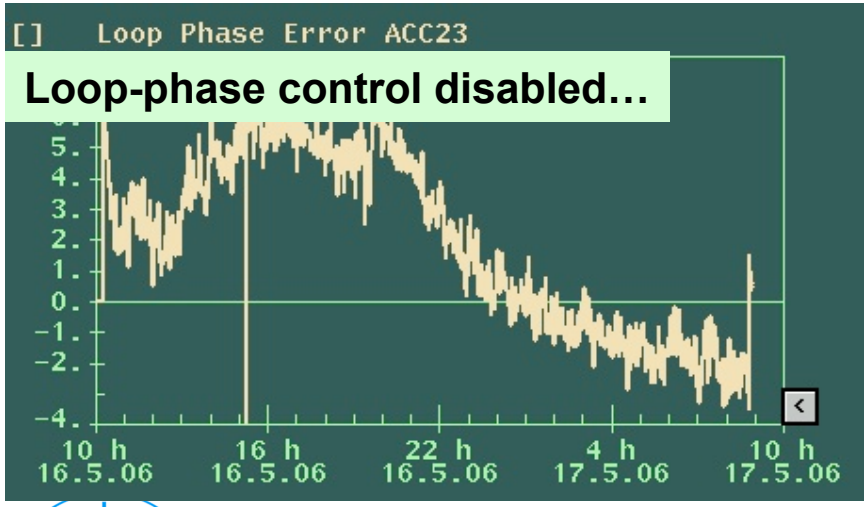
| | Configuration | Timeou | Short Description | Human-Readable Result | Numerical Res. | | |
|---|---|---|---|---|---|---|---|
| FUN0 | llrf/runDummy.csh out0.dat<br>llrf/out0.dat | 1500 | | Operation Successfull! | 0 | 1.96 | 0.28 |
| FUN1 | /home/da/doocs/solaris2/obj/server/ttf/klys/fsmv/runfsm<br>llrf/out1.dat | 500 | | Script execution error. | -99 | 0 | 0 |
| FUN2 | llrf/runOffCal.csh out2.dat<br>llrf/out2.dat | 1500 | Correcting Offsets. | Script execution error. | -99 | 0 | 0 |
| FUN3 | llrf/runSetSCandLP.sh out3.dat<br>llrf/out3.dat | 1500 | Correcting loopphase. | Corrected loop phase. | 2 | -3.84 | 0.12 |
| FUN4 | llrf/runOneStepAFFDSP.csh out4.dat<br>llrf/out4.dat | 1500 | Adapting Feedforward (DSP-par.). | Applied zero Feedforward (Feedforward was off) | 0 | 0 | 0 |
| FUN5 | llrf/runOneStepAFF.sh out5.dat<br>llrf/out5.dat | 1500 | Adapting Feedforward | Feedback is off. No FF-adaption possible. | 0 | 0 | 0 |
| FUN6 | llrf/runDefaultFF.csh out6.dat<br>llrf/out6.dat | 1500 | Applying default Feedforward. | Successfully set default feedforward. | 0 | 0 | 0 |
| FUN7 | llrf/runCouplerRst7.sh out7.dat 1<br>llrf/out7.dat | 1500 | Checking coupler interlocks. | Executing... | -98 | 0 | 0 |
| FUN8 | llrf/runSetSCandLP.sh out8.dat 1<br>llrf/out8.dat | 1500 | Checking loopphase. | No drive from llrf is applied. | 0 | 0 | 0 |
| FUN9 | llrf/runFieldQuality.sh out9.dat<br>llrf/out9.dat | 1500 | Checking data quality. | FB=0 or FF=0! Data quality is unacceptable. | 9 | inf | 117.42 |
| FUN10 | llrf/runCheckNetwork7.sh out10.dat<br>llrf/out10.dat | 1500 | Checking network. | Network works fine. | 0 | 0 | 0 |
| FUN11 | llrf/runRampFB.csh out11.dat<br>llrf/out11.dat | 1500 | Ramping Feedback. | Successfully ramped FB to target value! | 0 | 0 | 0 |
| FUN12 | llrf/runSetSPVal.csh out12.dat<br>llrf/out12.dat | 1500 | Setting amplitude and phase. | Successfully ramped SP to target value! | 0 | 0 | 0 |
| FUN13 | llrf/runOperatorAction.sh out13.dat<br>llrf/out13.dat | 1500 | Looking for operator action. | Executing... | -98 | 0 | 0 |
| FUN14 | llrf/runKathLaser.sh out14.dat<br>llrf/out14.dat | 1500 | Checking cathode laser settings. | Executing... | -98 | 0 | 0 |
| FUN15 | llrf/runGetDSPCalData.csh out15.dat<br>llrf/out15.dat | 1500 | Retrieving DSP calibration-data. | Script execution error. | -99 | 0 | 0 |
| FUN16 | llrf/runEvalDSPCalData7.csh out16.dat<br>llrf/out16.dat | 1500 | Evaluating DSP-Cal-Data. | Script execution error. | -99 | 0 | 0 |
| FUN17 | llrf/runSignalCalib7.sh out17.dat<br>llrf/out17.dat | 1500 | Calculating cavity signal calibration. | There is no drive at all! | 2 | 0 | 0 |
| FUN18 | llrf/runGetDetunings.csh out18.dat<br>llrf/out18.dat | 1500 | Calculating detunings. | Calculating detuning done. | 0 | 0 | 0 |
| FUN19 | llrf/runSetSPVal.csh out19.dat<br>llrf/out19.dat | 1500 | Setting amplitude and phase (slowly). | | 0 | 0 | 0 |
| FUN20 | llrf/runKlyIsDown.sh out20.dat<br>llrf/out20.dat | 1500 | Looking if Klystron is alive. | Executing... | -98 | 0 | 0 |
| FUN21 | llrf/runFindFancyPulses.sh out21.dat<br>llrf/out21.dat | 1500 | Looking for fancy pulses. | Feedforward is off. | 0 | 0 | 0 |
| FUN22 | offCal ACC456 out22.dat<br>out22.dat | 0 | | Script execution error. | -99 | 0 | 0 |
| FUN23 | setSCandLP ACC456 out23.dat<br>out23.dat | 0 | | Calculated bandwidth out of range. (Maybe there is no drive signal at all?) | -111 | 0 | 0 |
| FUN24 | llrf/runRippleLoad.sh out24.dat<br>llrf/out24.dat | 500 | Applying ripple-correction. | Script execution error. | -99 | 0 | 0 |
| FUN25 | llrf/runRippleLoad.sh out25.dat 0<br>llrf/out25.dat | 500 | Restoring zero ripple-table. | Script execution error. | -99 | 0 | 0 |
| FUN26 | defaultFF ACC456 out26.dat<br>out26.dat | 0 | | Script execution error. | -99 | 0 | 0 |
| FUN27 | | 0 | | | 0 | 0 | 0 |
| FUN28 | ./runfsmklys4 out28.txt<br>out28.txt | 0 | | Script execution error. | -99 | 0 | 0 |
| FUN29 | ./runfsmklys5 out29.txt<br>out29.txt | 0 | | Script execution error. | -99 | 0 | 0 |

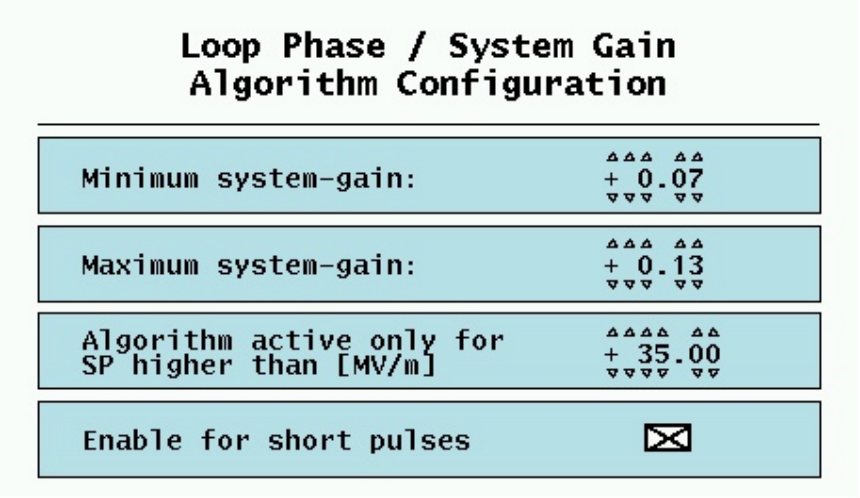
# Adaptive Feedforward



# Loop Phase Correction



# Off-Line Restructurability

emacs@

File Edit Options Buffers Tools C++ Help

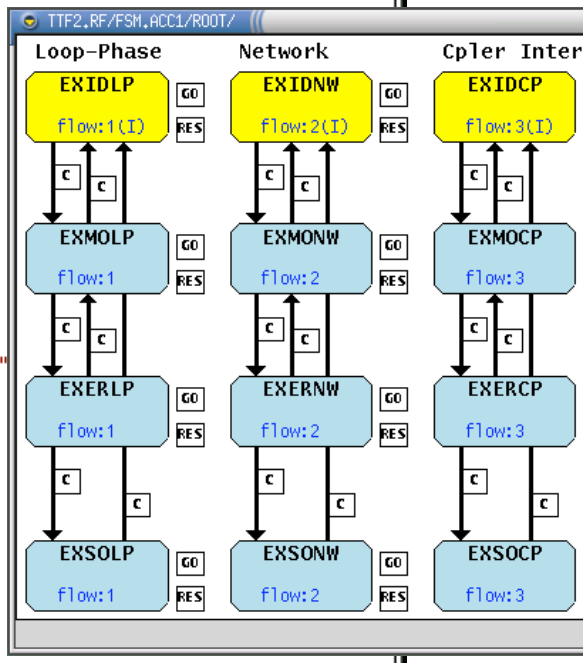
```
//// DO NOT EDIT THIS AND NEXT TWO ROWS
// init function for state 'EXERDQ'
int     ROOT_EXERDQ_init_action(FSMstate *st_p_)
{

        fputs("ROOT_EXERDQ:: \"init\" event  procedure: ROOT_EXERDQ_init_action \n", stderr);

        TStateLogic* sd=TStateLogic::factory("EXERDQ");
        sd->set_go("ROOT_EXCEPTION.EXERDQ_GO");
        sd->set_res("ROOT_EXCEPTION.EXERDQ_RES");
        sd->add_event("EXERDQ", "exerdq_exmodq", "ROOT_EXCEPTION.EXERDQ_T1");
        sd->add_event("EXERDQ", "exerdq_exsodq", "ROOT_EXCEPTION.EXERDQ_T2");

        return ST_OK;
}

//// DO NOT EDIT THIS AND NEXT TWO ROWS
// enter function for state 'EXERDQ'
int     ROOT_EXERDQ_enter_action(FSMstate *st_p_)
{

        fputs("ROOT_EXERDQ:: \"enter\" event  procedure: ROOT_EXERDQ_enter_action \n"

        TStateLogic* sd=TStateLogic::factory("EXERDQ");
        sd->enter(st_p_);

        return ST_OK;
}

//// DO NOT EDIT THIS AND NEXT TWO ROWS
// during function for state 'EXERDQ'
int     ROOT_EXERDQ_during_action(FSMstate *st_p_)
{

        fputs("ROOT_EXERDQ:: \"during\" event  procedure: ROOT_EXERDQ_during_action \n", stderr);

        TStateLogic* sd=TStateLogic::factory("EXERDQ");
        sd->during(st_p_);

        return ST_OK;
}
```

*init*

*enter*

*during*

--:** llrf_fsm_panel_ROOT.cc (C++ Abbrev)--L1949--94%

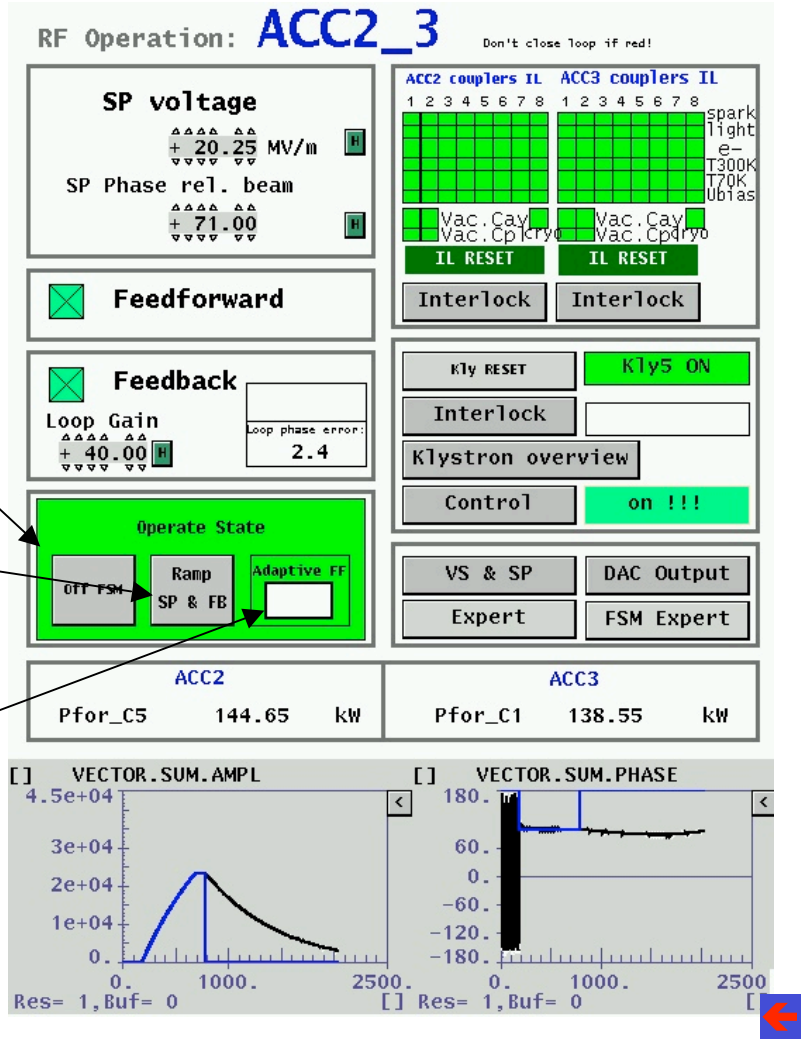


# Simple Operators View

**Switch off FSM**

**Ramp RF to the target values (…)**

**Enable adaptive FF**



# FSM-Expert View

Checkboxes determine the behavior of the algorithms (and the algorithms determine the behavior of the FSM).

Scrolling messages from all procedures

Debug level

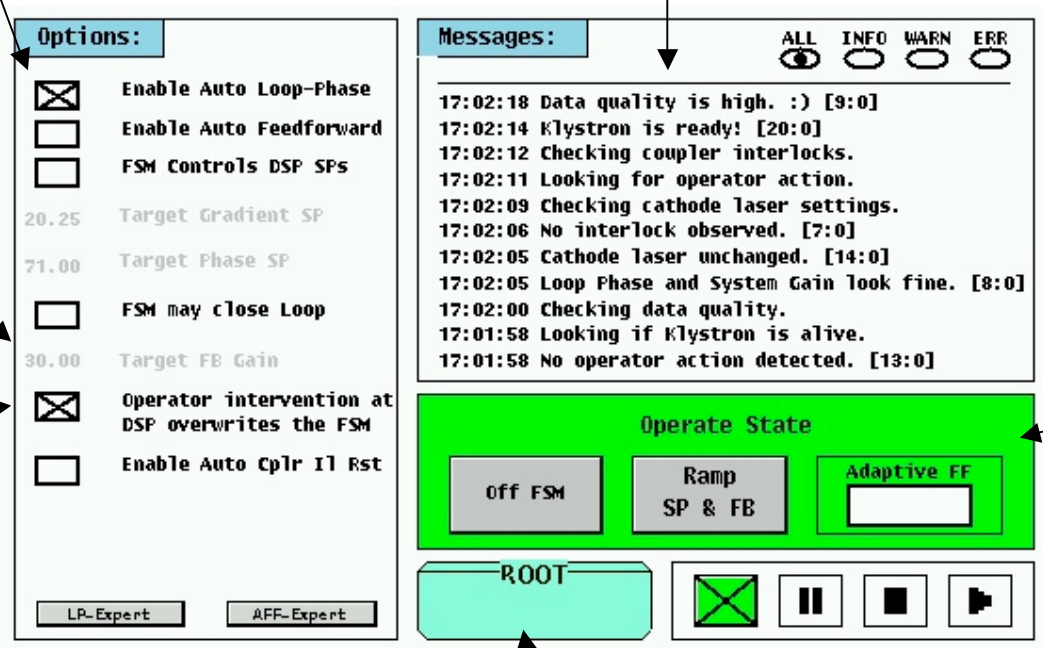
Target values are automagically updated while operator is working with the DSP!

…but only as long as this is desired!

Simple operators view

Algorithm expert settings

Access to FSM-structure (for on-line reconfig)



# DDD FSM Framework

- Nice, but way to unflexible
- That's why my implementation extended it by
  - Procedure server (with many features like timeout and email)
  - Factory-classes for a simple reconfiguration (still needs compilation)
  - Parser that interprete online configuration ("go", "res" and "c" - buttons)
- Wish: have restructuration of states on the fly!

# Overview of Hera Automation

DESY LLRF Automation Workshop

S. Herb, 22.05.06

# Instant Summary

- Hera is ***not*** a one-button machine !!

- But partial automation is crucial

Þ "**Interactive Automation**"

Operators and Automation systems must work smoothly together ('*mix of human, software agents*')

# Hera Automation History

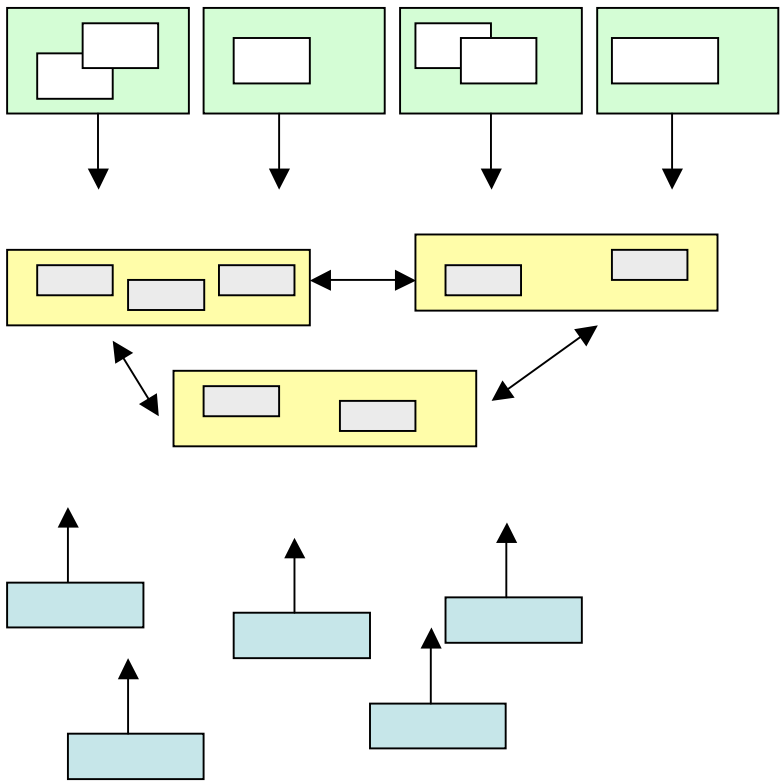
0. Start with many Operators sitting at many Consoles

1. Try to replace operator tasks by console level software (lots of graphics, operator interactivity)

2. Try to migrate software tasks from console to ‘middle layer’ (‘less’ interactivity)

=> Naturally described by multiple Agents

# Control System Structure



Console Programs (**Clients**)
don't talk to each other !!

Middle Layer (**Client/Server**)

Shared Services

Integration Layer

Hardware Control Layer:
Low level **Servers** avoid
talking to each other

Automation (of multiple devices) **belongs** in the middle layer

But it isn't always easy

# Automation *Systems* (Selection)

- p momentum/tune/chromaticity control
- e orbit stabilizer
- Sequencer { return success_story; }
- Permit Server { return not_success_story; }
- Optics measurements ( VB, Matlab)

# Why has Automation been difficult?

- Complicated Machine, heterogeneity
- Complicated sequences, frequent changes
- Incomplete / unreliable beam diagnostics
- Dynamic error handling non-trivial
- Control System heterogeneity (efficiency)
- Inadequate low-level encapsulation
- Inadequate operations maintenance
- Hard to get beam-time for development/testing

# SNS (Linac RF) Automation

**May 2006**

**Kay-Uwe Kasemir**

**kasemirk@ornl.gov**

# SNS Linac

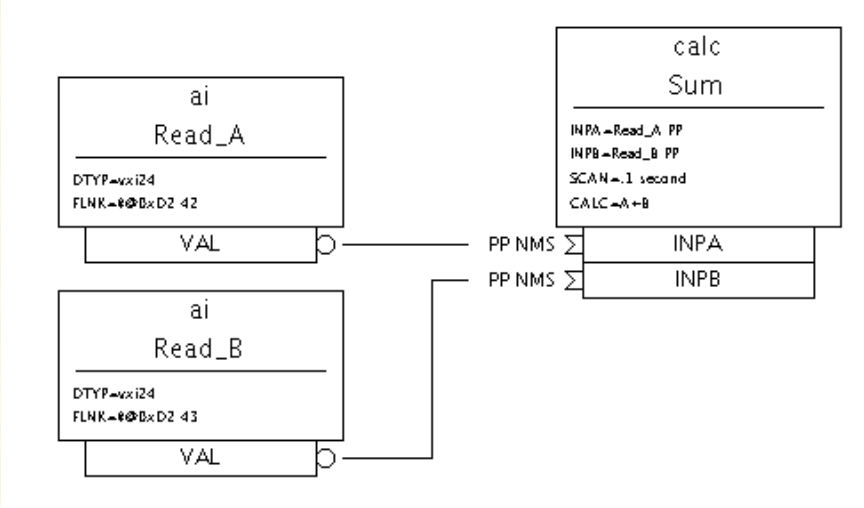
- 96 Cavities
  - 15 warm, rest superconducting.
- One LLRF System per Cavity
  - Run by 55 LLRF front-end computers (EPICS “input-output-controllers”, IOCs)
- One HPRF Klystron per Cavity
  - Exception: 4 MEBT cavities have power amplifier.
  - High-Voltage Modulators, providing the klystron voltage, are shared by 2 …12 cavities.
  - Total of 11 HPRF IOCs.

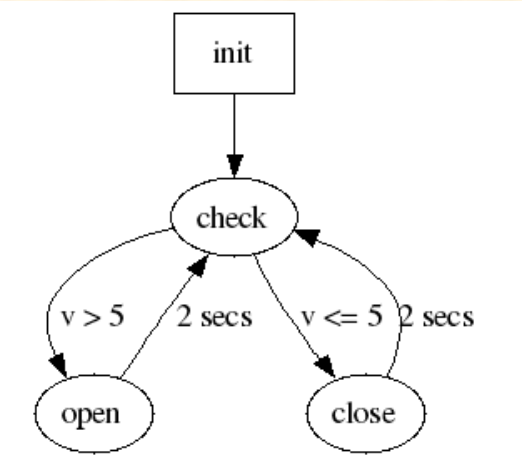
# Define "Automation"

- PID controller implemented via OP-Amps, interlocks via NAND gates and monoflops?
  - Personnel protection system uses some of that.
- … implemented in FPGA or DSP?
  - SNS LLRF uses that.
- Interlocks and remote control via PLCs?
  - SNS conventional facilities, water cooling, cryo plant, vacuum, HPRF, machine & personnel protection use that a lot.
- Control System Perspective:
  - All the above are "given"; black-box systems.

# Control System Perspective

- **Added in the EPICS IOC:**
  - **C/C++ driver code to interface custom devices.**
  - **EPICS real-time database logic**
    - **Periodic processing, scanned or event driven.**
    - **Mostly "steady state", fast and repetitive.**
    - **Provides network access to data in FPGA & PLC.**
  - **Sequences (FSMs)**
    - **Transitions between states; timeouts; different modes of operation.**
    - **Slower, many conditionals .**
    - **Automation added on top of FLGA, PLC, database.**

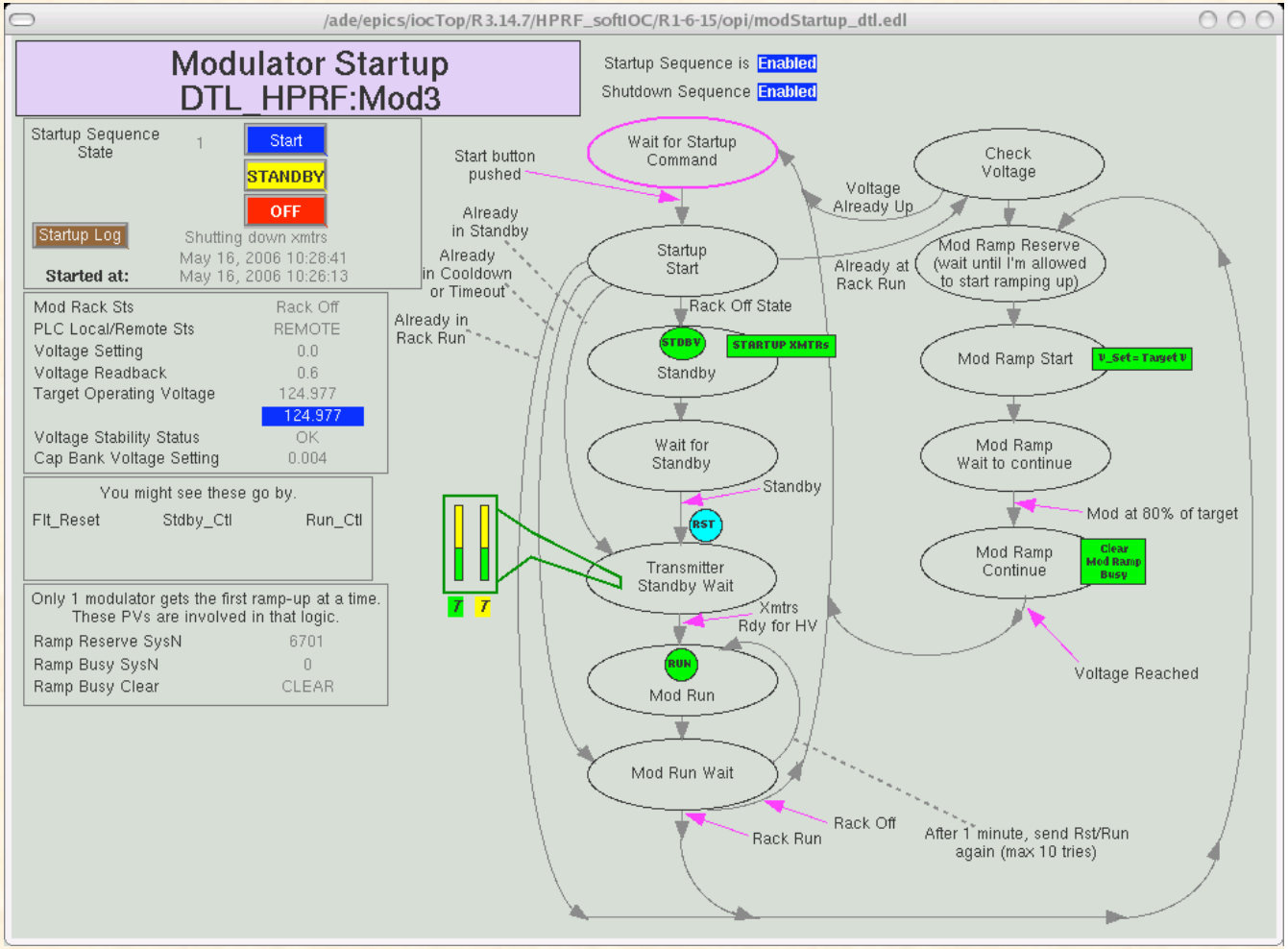


# Automation Example: High-Voltage Modulator Startup

**(Pam Gurd, ORNL)**

- **Initially…**
  - **performed via lower-level operator screens, directly interacting with the real-time database, maybe based on written recipes or cheat-sheets.**
  - **OK while only first few modulators were on-line.**
- **Sequencer**
  - ***automates* this tedious task,**
  - **reduces errors,**
  - **speeds up downtime recovery.**

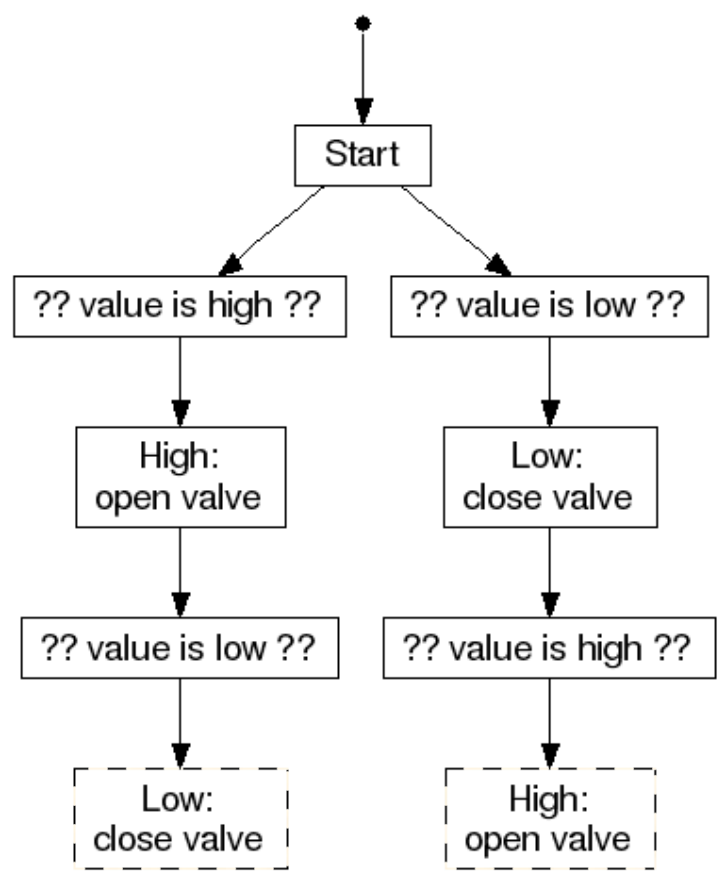


# Specifications…

- "Based on what we saw last night, you need to stop the RF when XXX trips"
  - What does XXX "OK" vs. "tripped" mean? "HPRF OK" check turned into another sequence:
    - Transmitter in "Ready for HV" Mode.
    - Modulator in "Rack Run" Mode.
    - ~~Modulator voltage within 1% of its setpoint?~~ Modulator ignores its setpoint!
    - Settle for >50kV, stable within 1% for >10 secs.
  - "Stop the RF" as in
    - Reduce amplitude?
    - Reduce to 0?
    - Recover when trip condition clears, or stay off?
  - What overrides are needed? Who's allowed to override?
  - How to notify operators of the original reason, and prevent being the shot messenger?

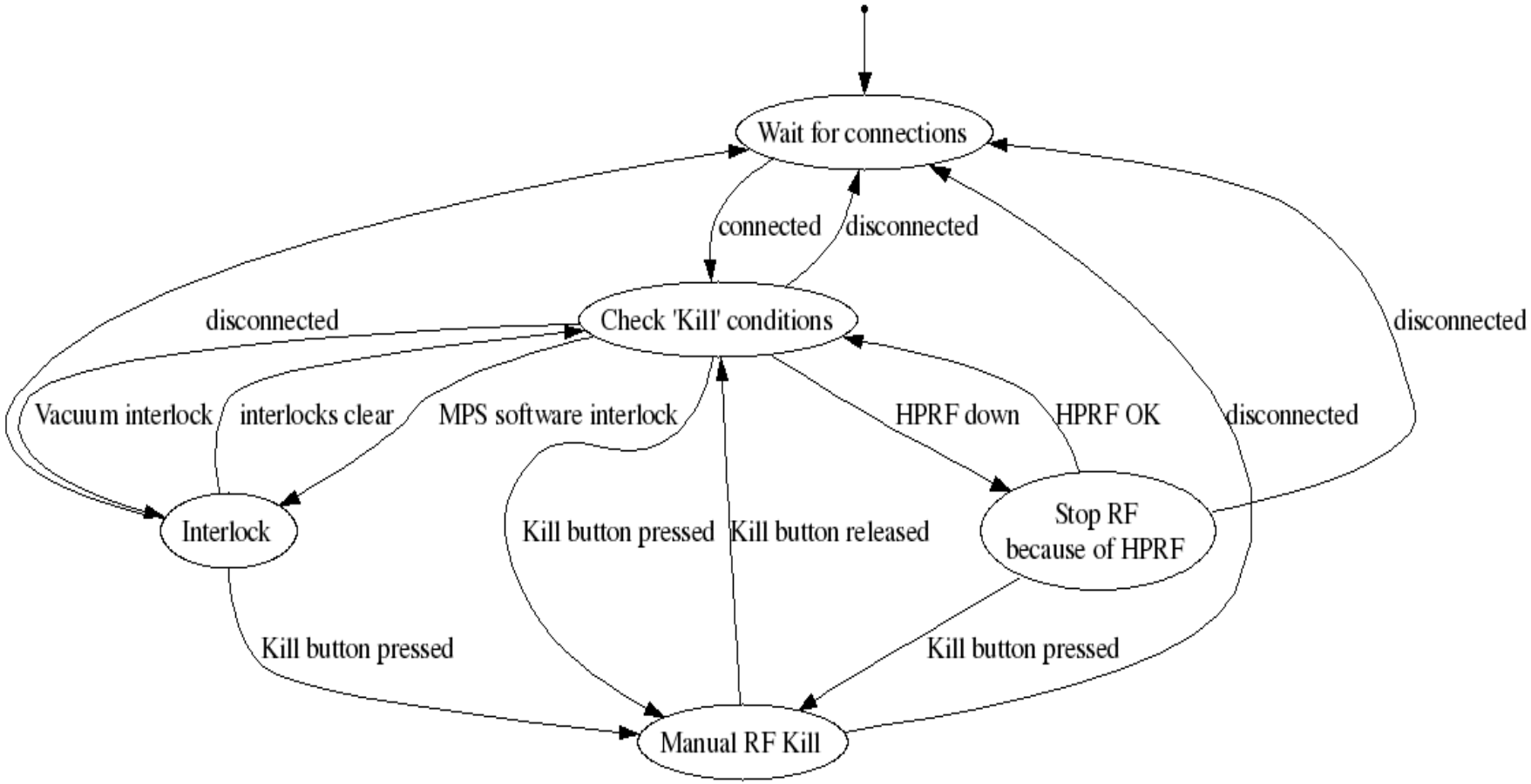
# Dromey Behavior Tree

- Dromey (http://www.sqi.gu.edu.au/gse) developed a mechanism for handling complex requirements in "Behavior Tree" notation.
- Carl Lionberger (LBL) developed some EPICS sequence code based on DBTs in a manual process.
- One neat idea:
  Show that a certain state is entered again without drawing the transition "back" to the original state.



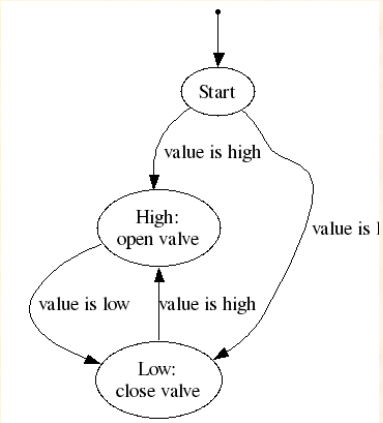
# Another Real-World State Diagram

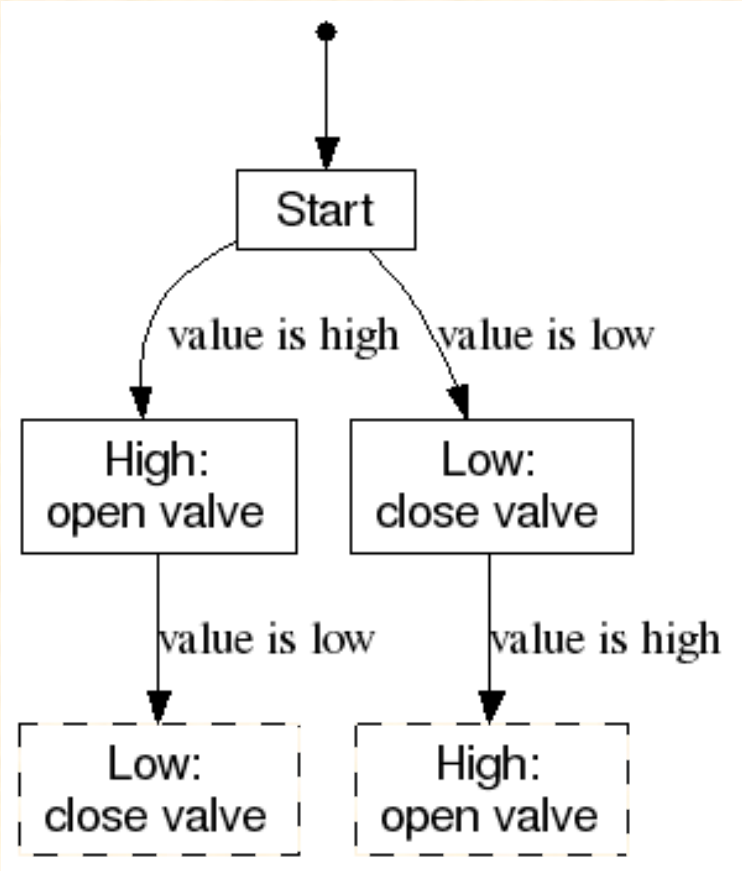
- **Even "simple" FSM complicated by**
  - **Error conditions ("disconn.")**
  - **Manual interaction.**



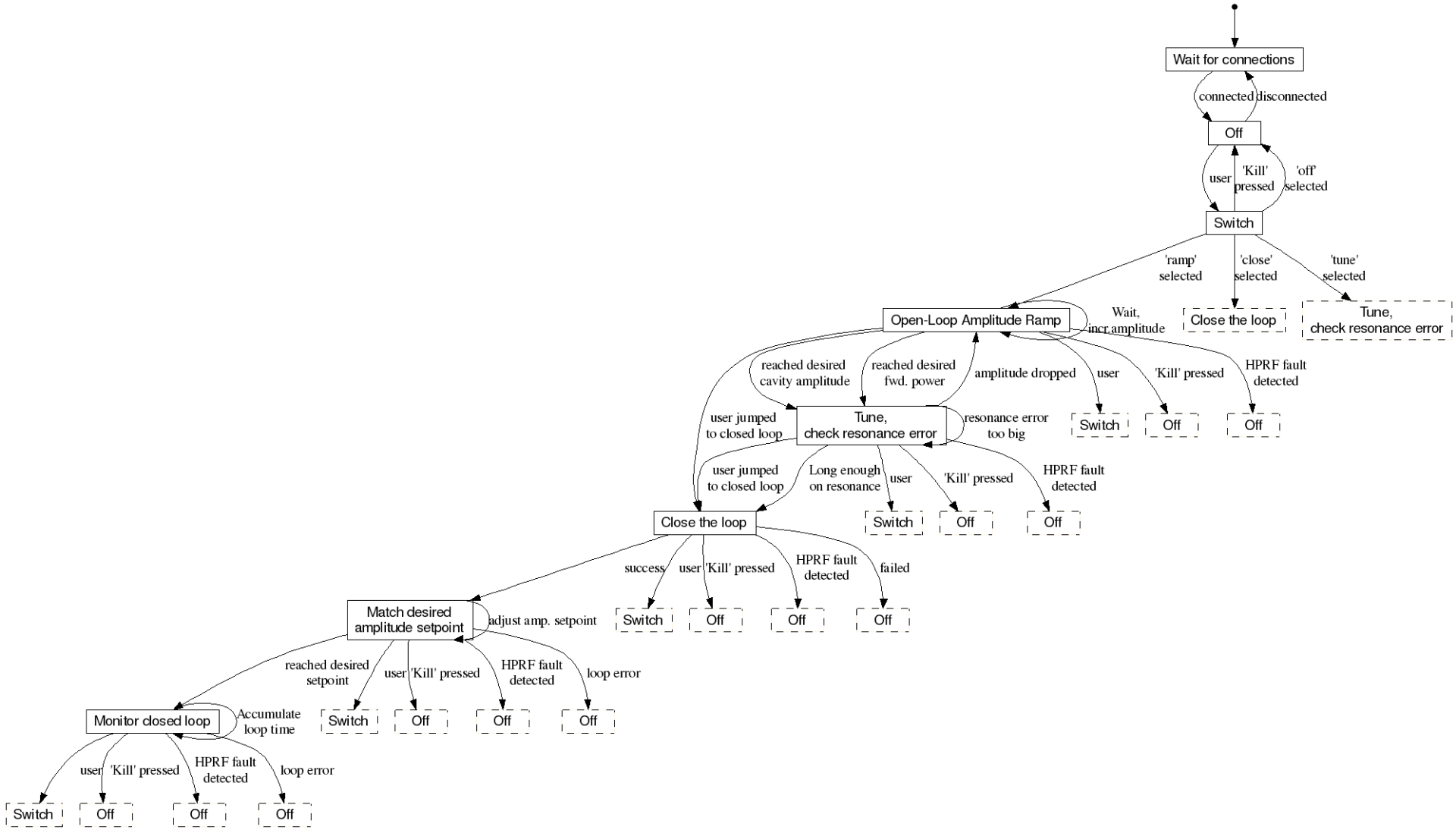
# Mix of ideas

- Basically the old state diagram, but with Dromey's idea for representing cycles.
- Developing a simple tool that can create
  a) Original state diagram
  b) Dromey Behavior Tree
  c) The mix shown here
  <u>from the original sequencer source file</u>, aided by special comments in the code.



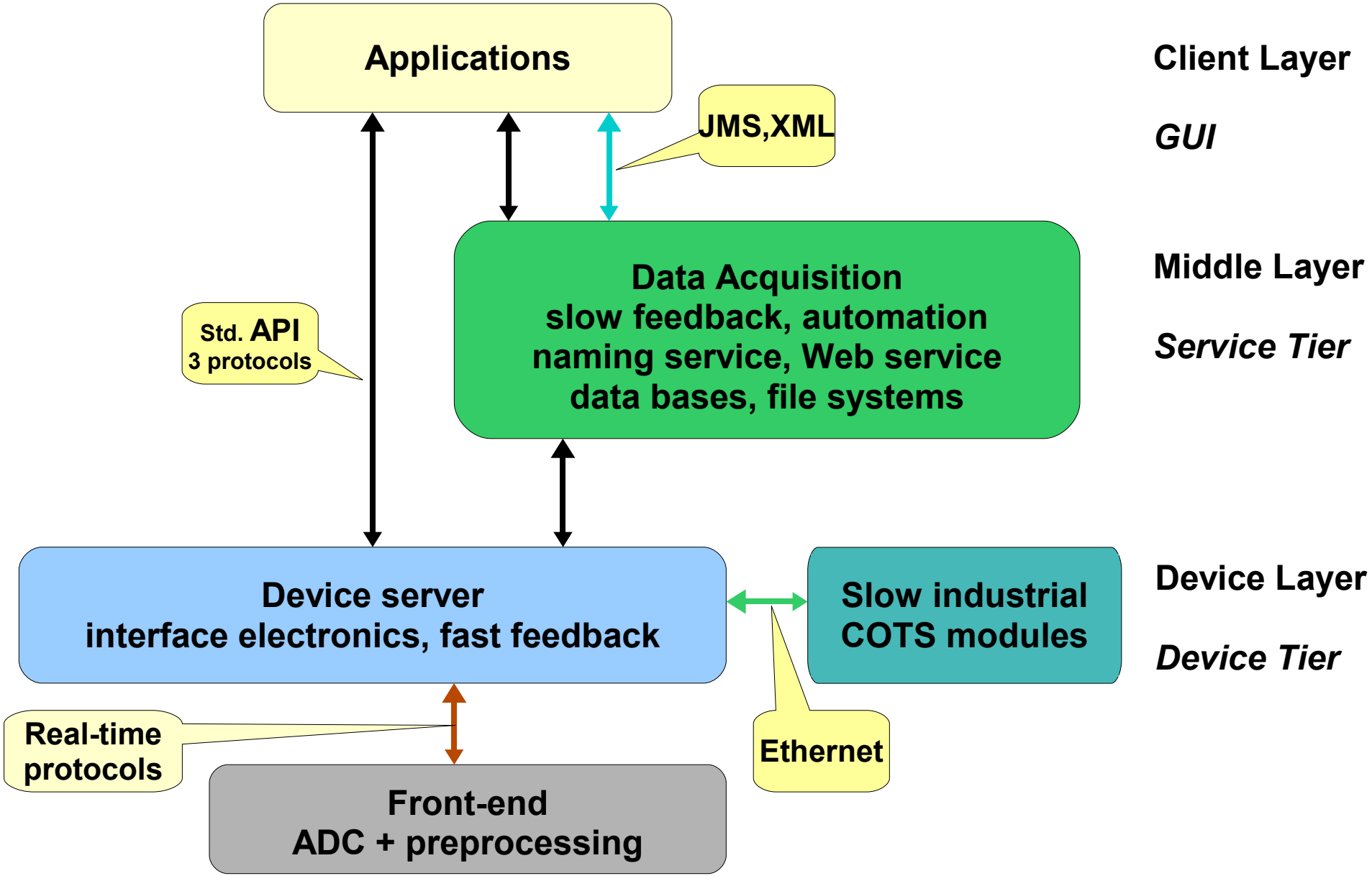
# .. and Mixed Representation



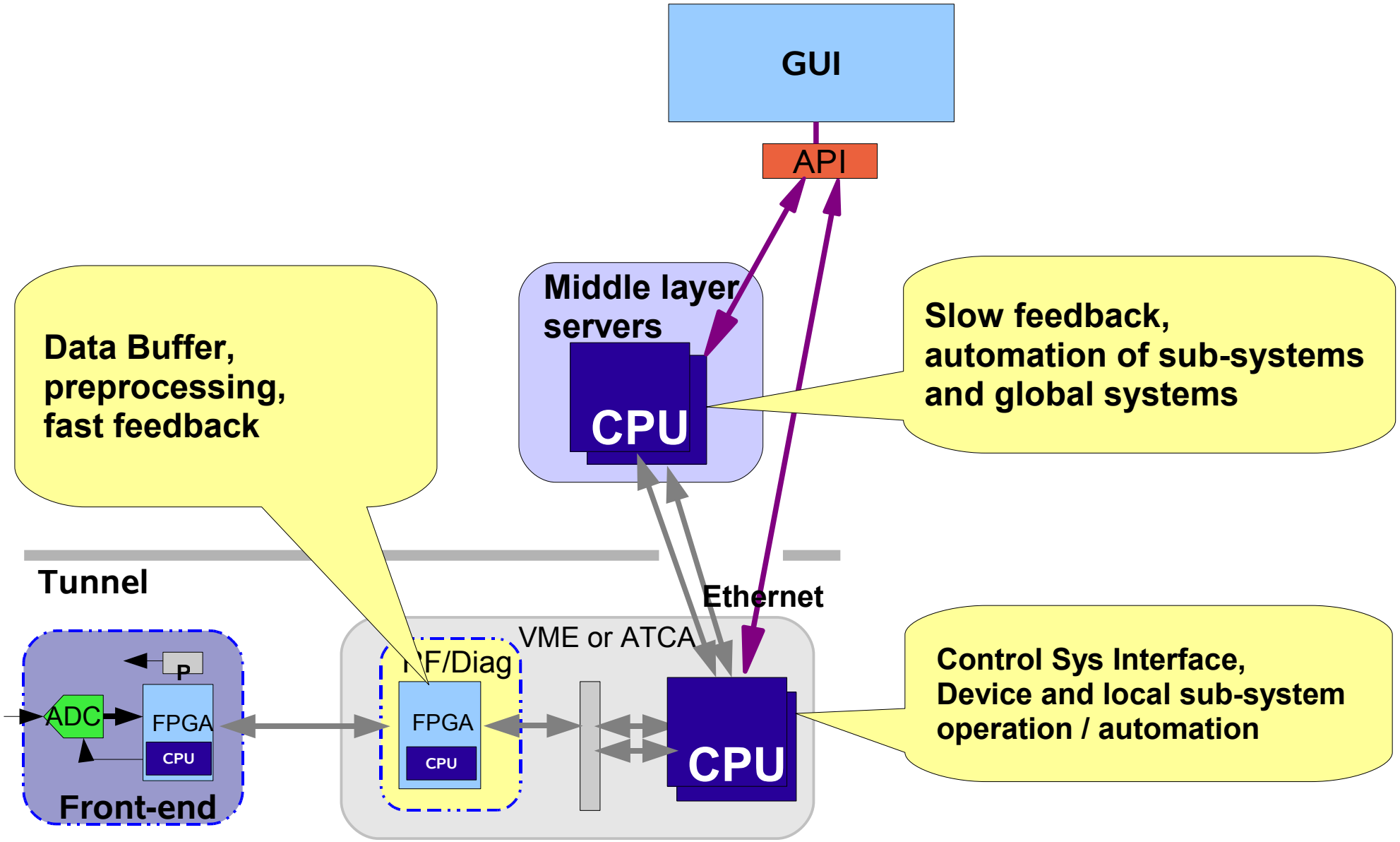
# DOOCS and Automation

**Kay Rehlich**
for the DOOCS Team

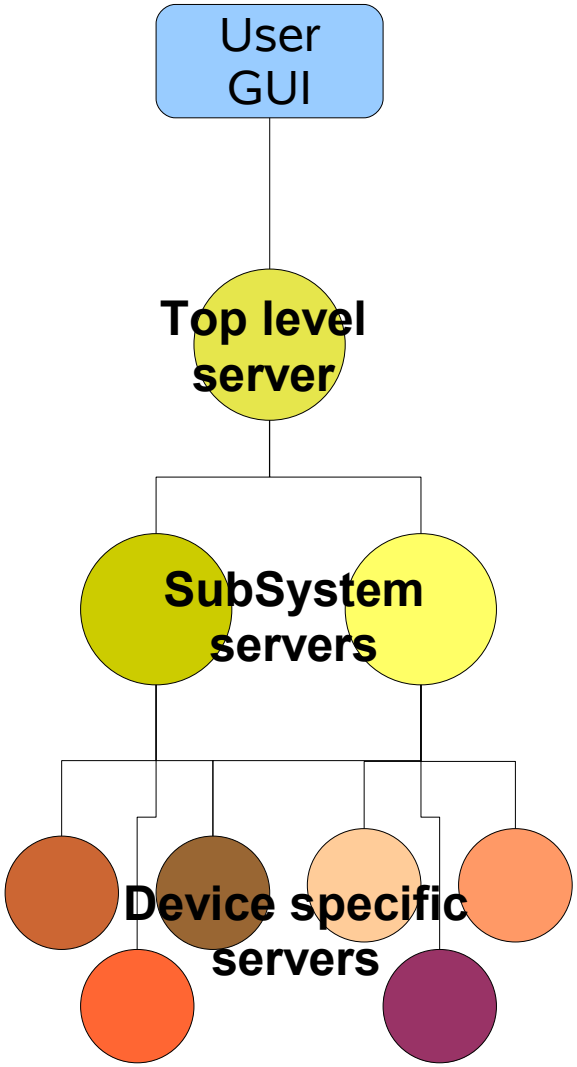
# Architecture



# Data Processing

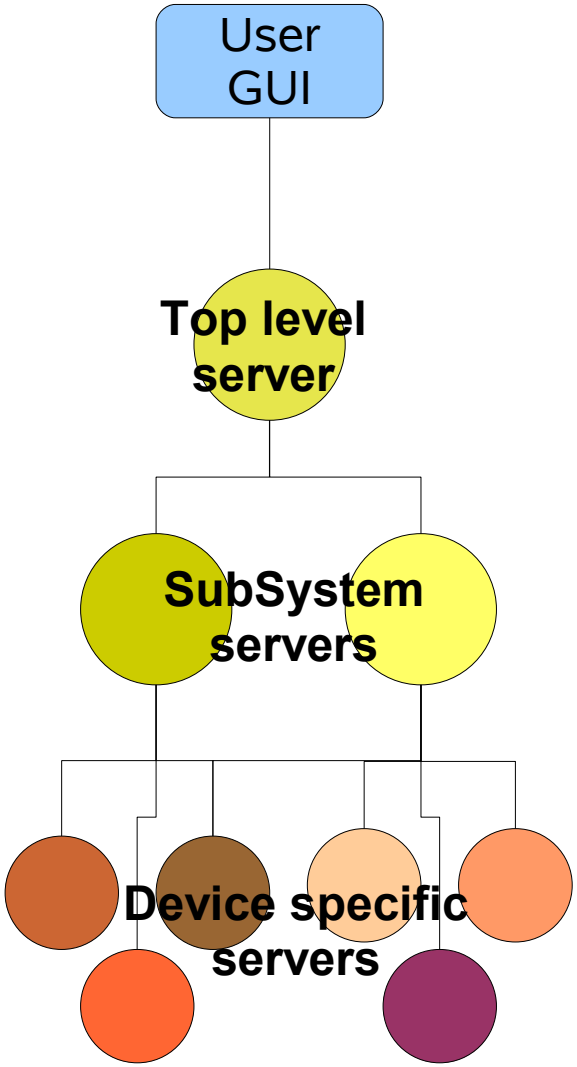


# Automation



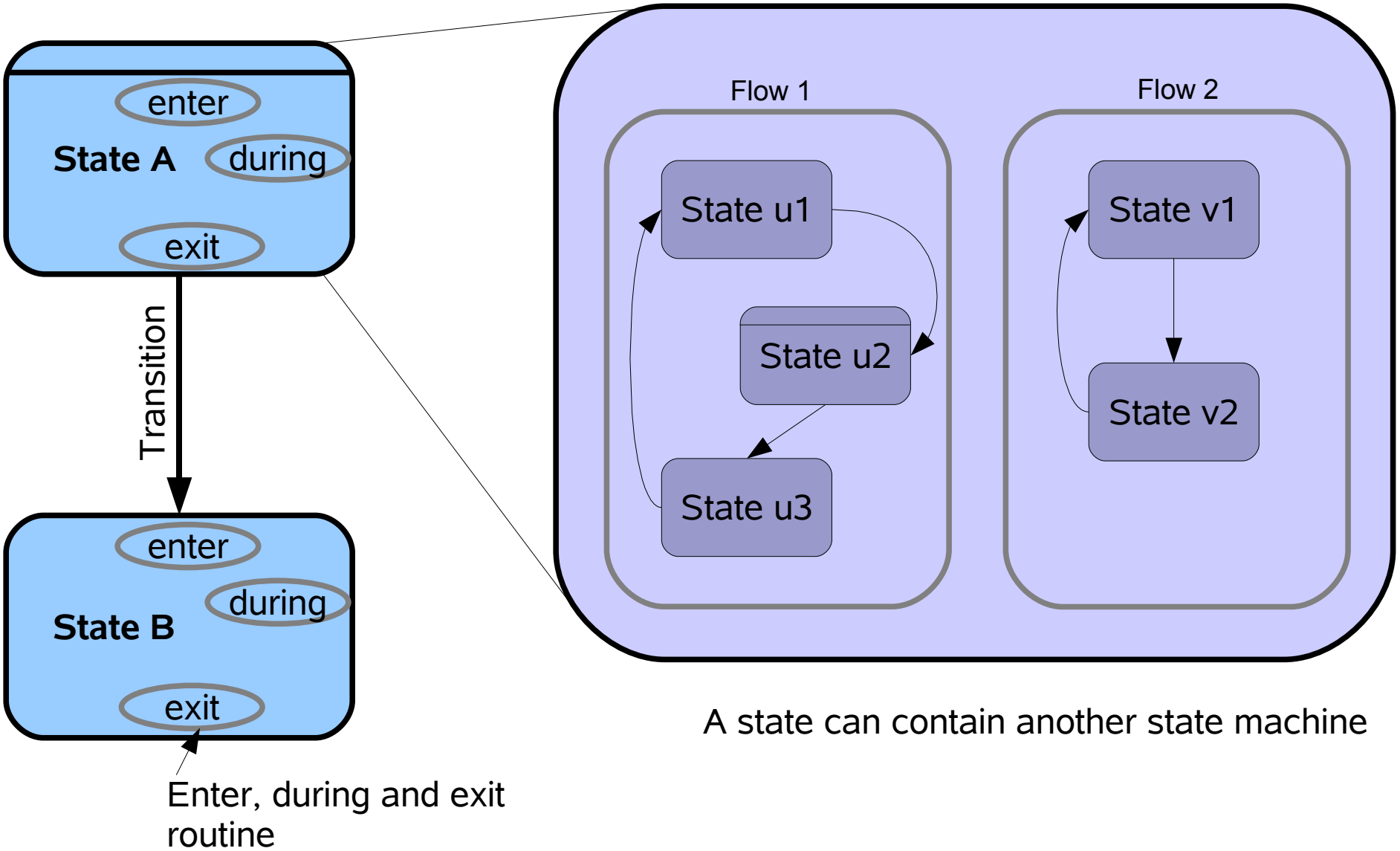

- Hierarchical structure
  - Grouping of functions
  - To reduce the complexity
  - To keep devices independent
  - To define simple and common inter-faces
- Finite State Machines and processes
  - Integration of high level tools (e.g. MATLAB procedures)
- Full control system integration
- Debug and maintenance provisioning

# Automation

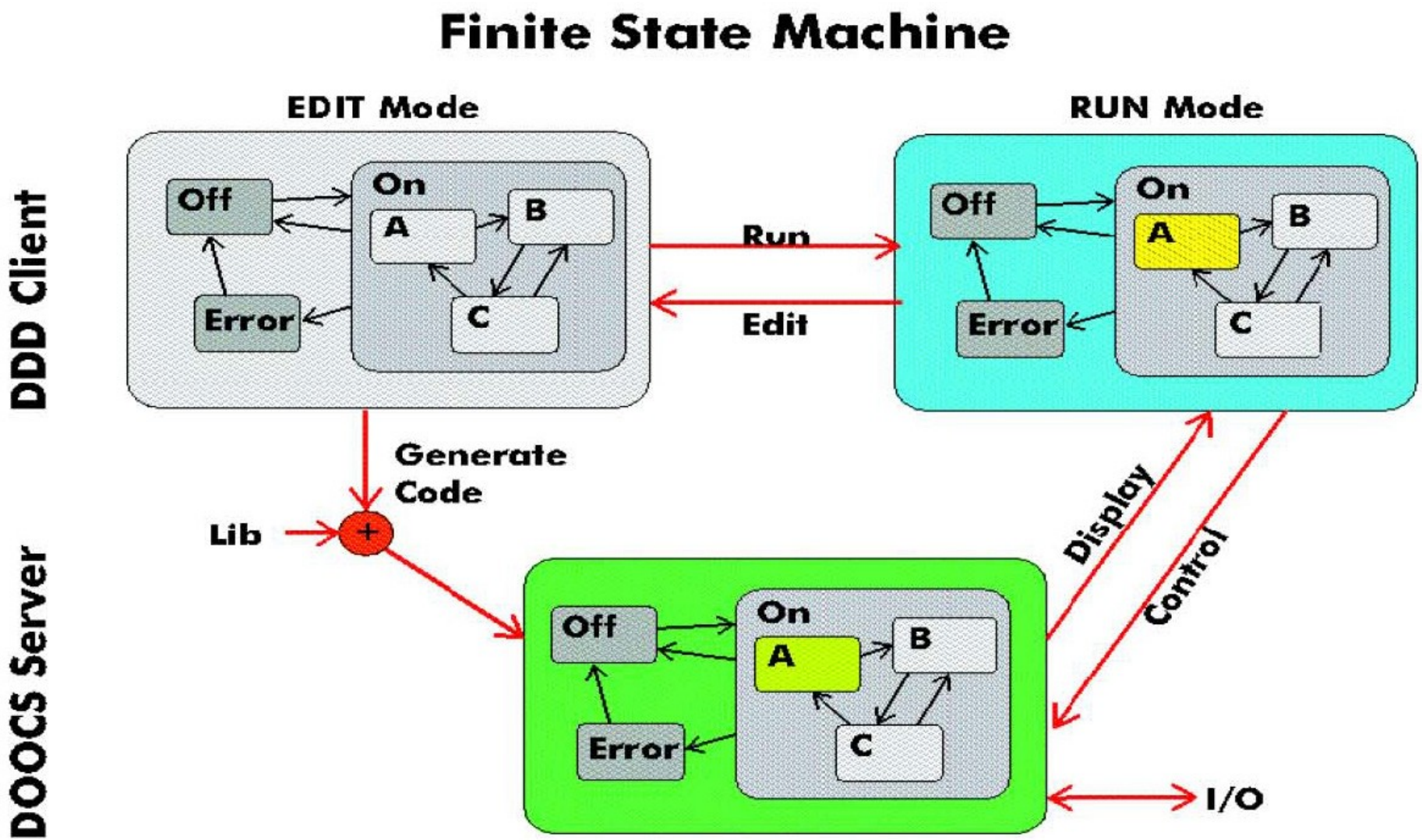



- Hierarchical structure
  - Grouping of functions
  - To reduce the complexity
  - To keep devices independent
  - To define simple and common inter-faces
- Finite State Machines and processes
  - Integration of high level tools (e.g. MATLAB procedures)
- Full control system integration
- Debug and maintenance provisioning

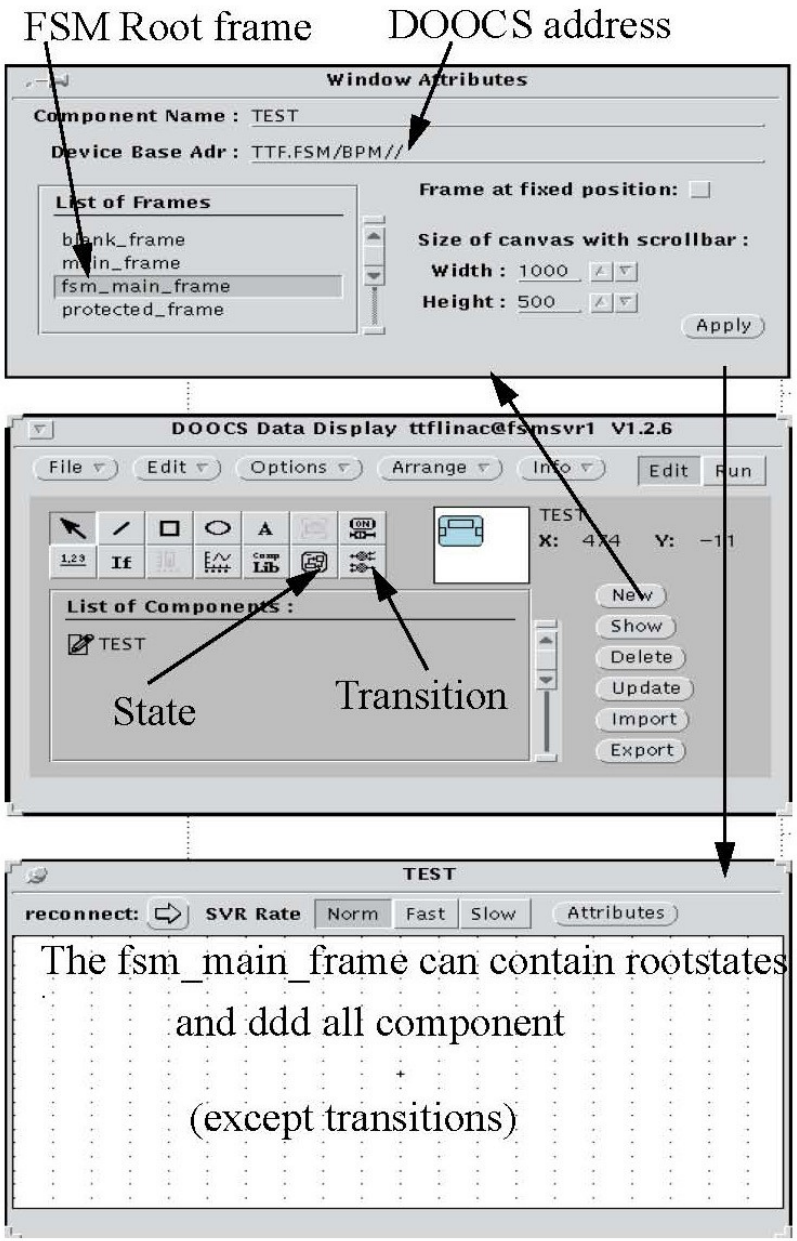
# States, super states, flows, transitions

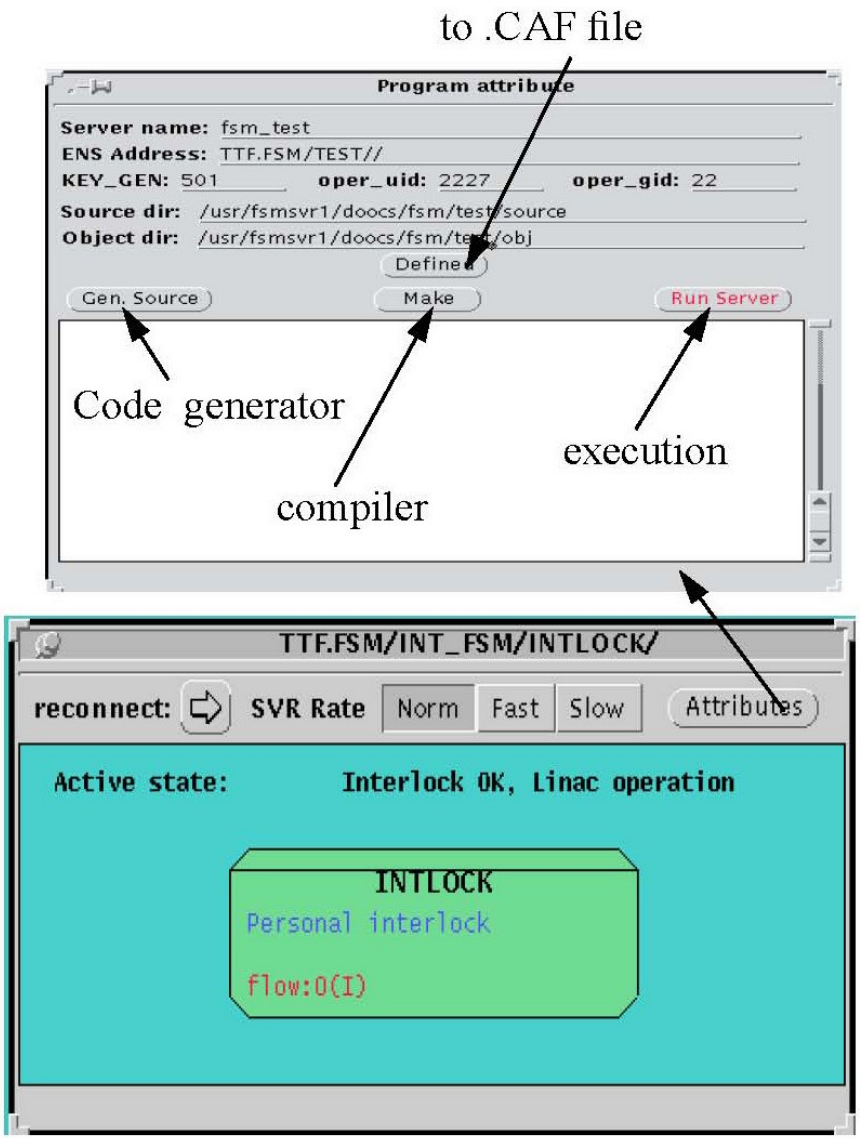


# FSM Design

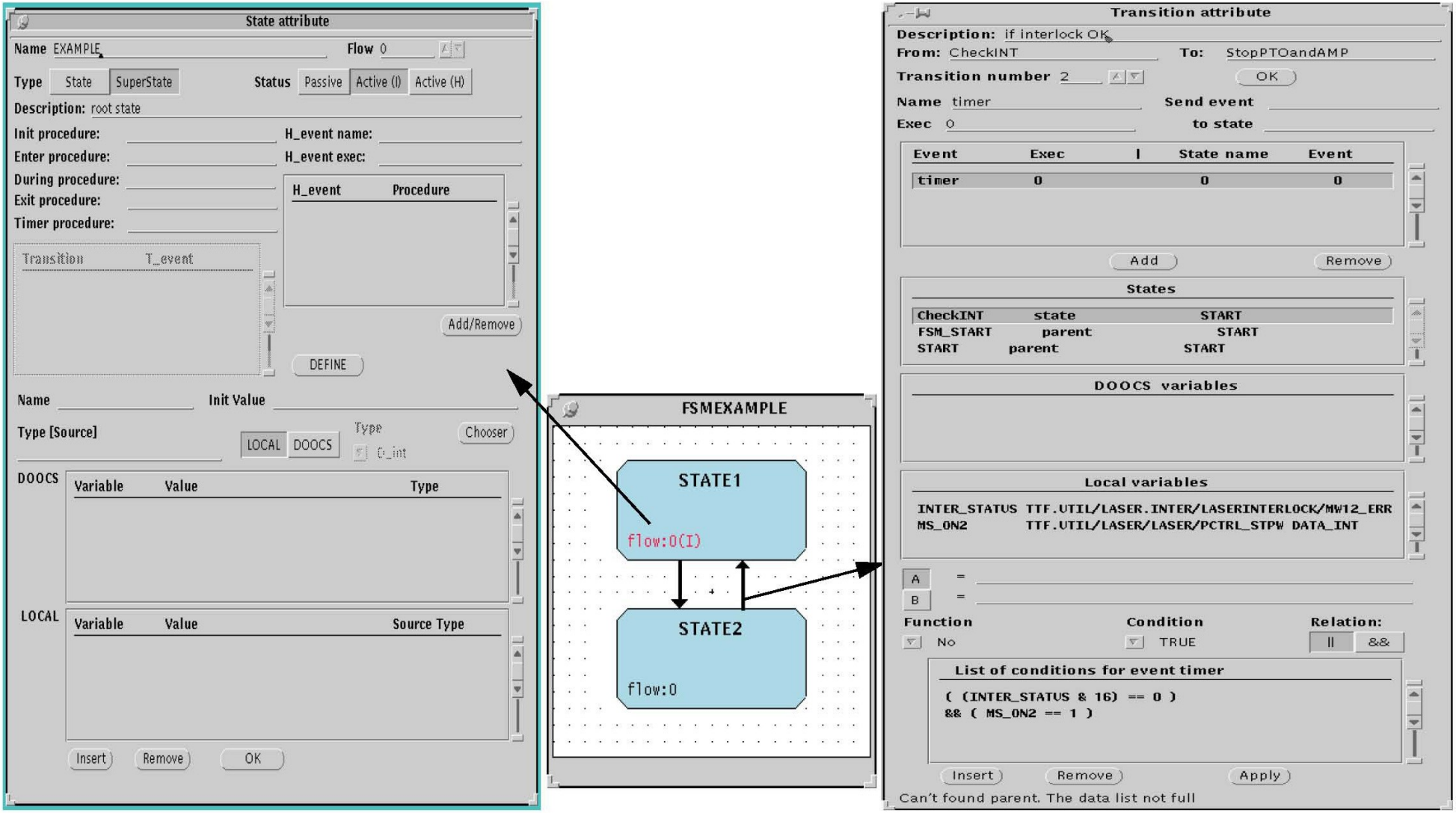

Finite State Machine
EDIT Mode
RUN Mode
DDD Client
DOOCS Server
Off
On
A
B
C
Error
Run
Edit
Generate Code
Lib
+
Display
Control
I/O

# FSM Creation in ddd



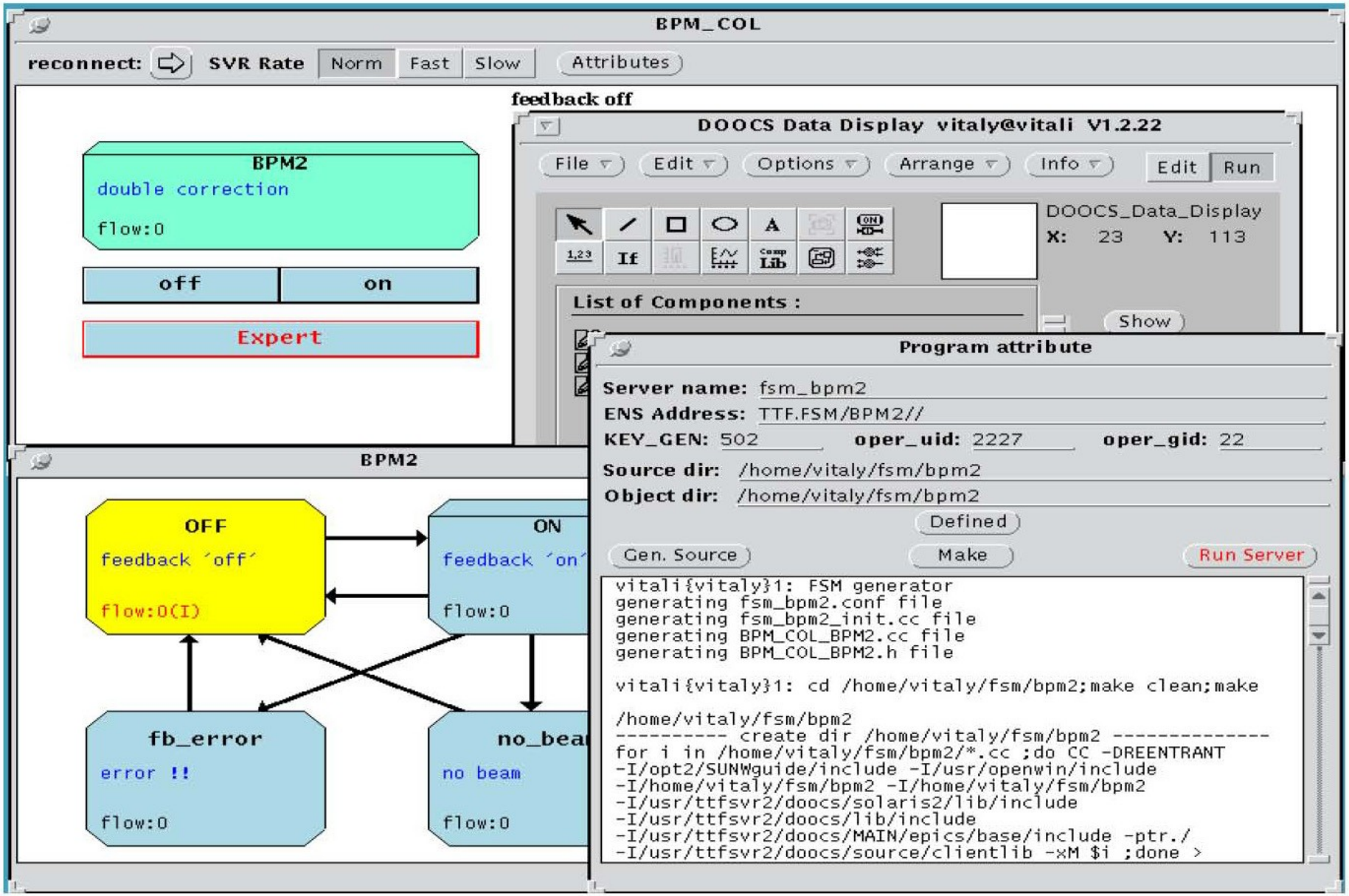
# 'State' and 'Transition' Panel



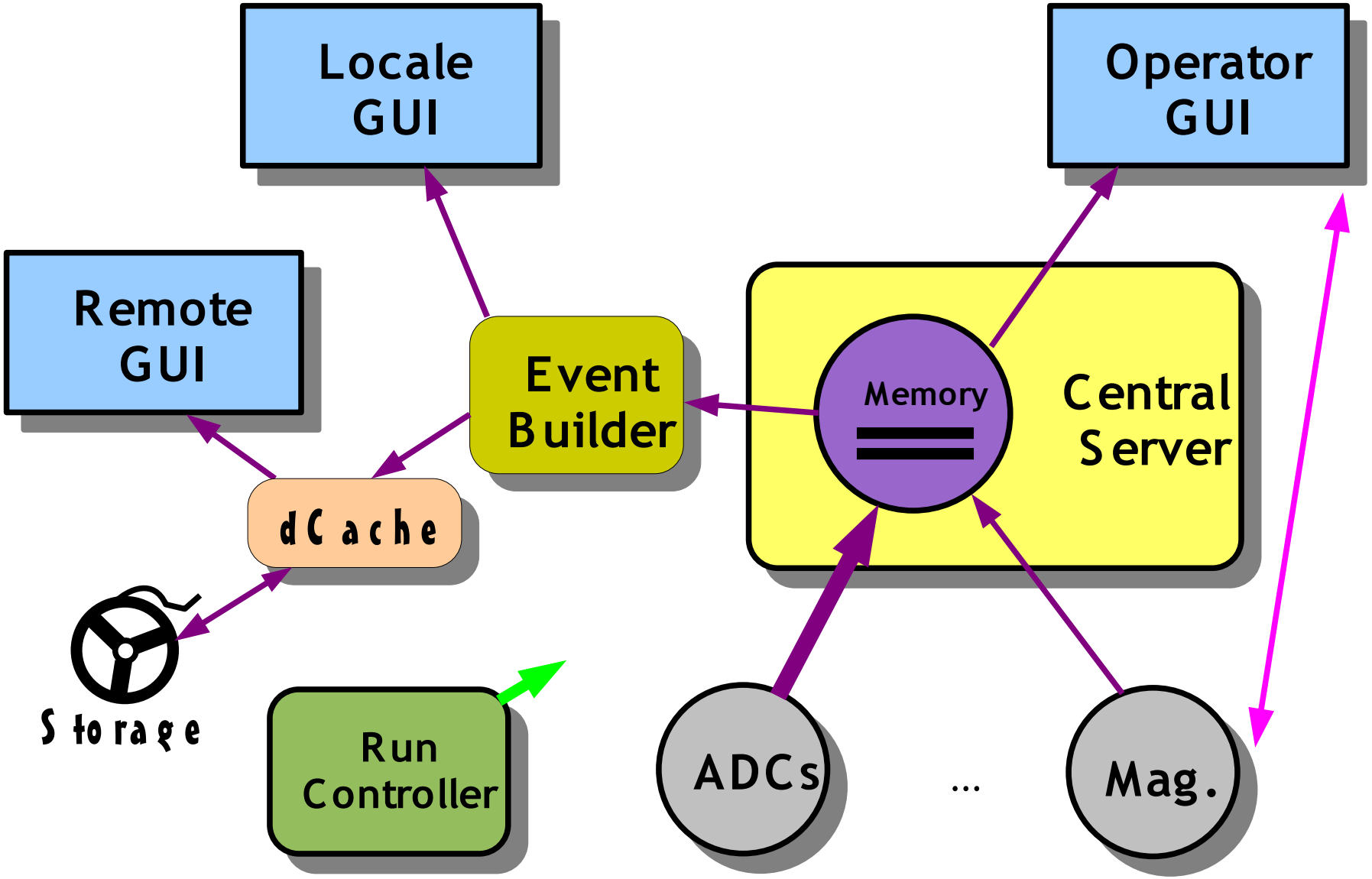
# C++ Code to Add

```
//F.S.M Library file

#include "fsm.h"

//// DO NOT EDIT THIS AND NEXT TWO ROWS
// during function for state 'BP_FB'
intBP_FB_BP_FB_during_bp_fb_dur(FSMstate *st_p_)
{
  puts("BP_FB_BP_FB:: 'during' event  procedure: BP_FB_BP_FB_during_bp_fb_dur \n", stderr);
  return ST_OK;
}

//// DO NOT EDIT THIS AND NEXT TWO ROWS
// exit function for state 'BP_FB'
intBP_FB_BP_FB_exit_bp_fb_ex(FSMstate *st_p_)
{
  fputs("BP_FB_BP_FB:: 'exit' event  procedure: BP_FB_BP_FB_exit_bp_fb_ex \n", stderr);
  return ST_OK;
}
```

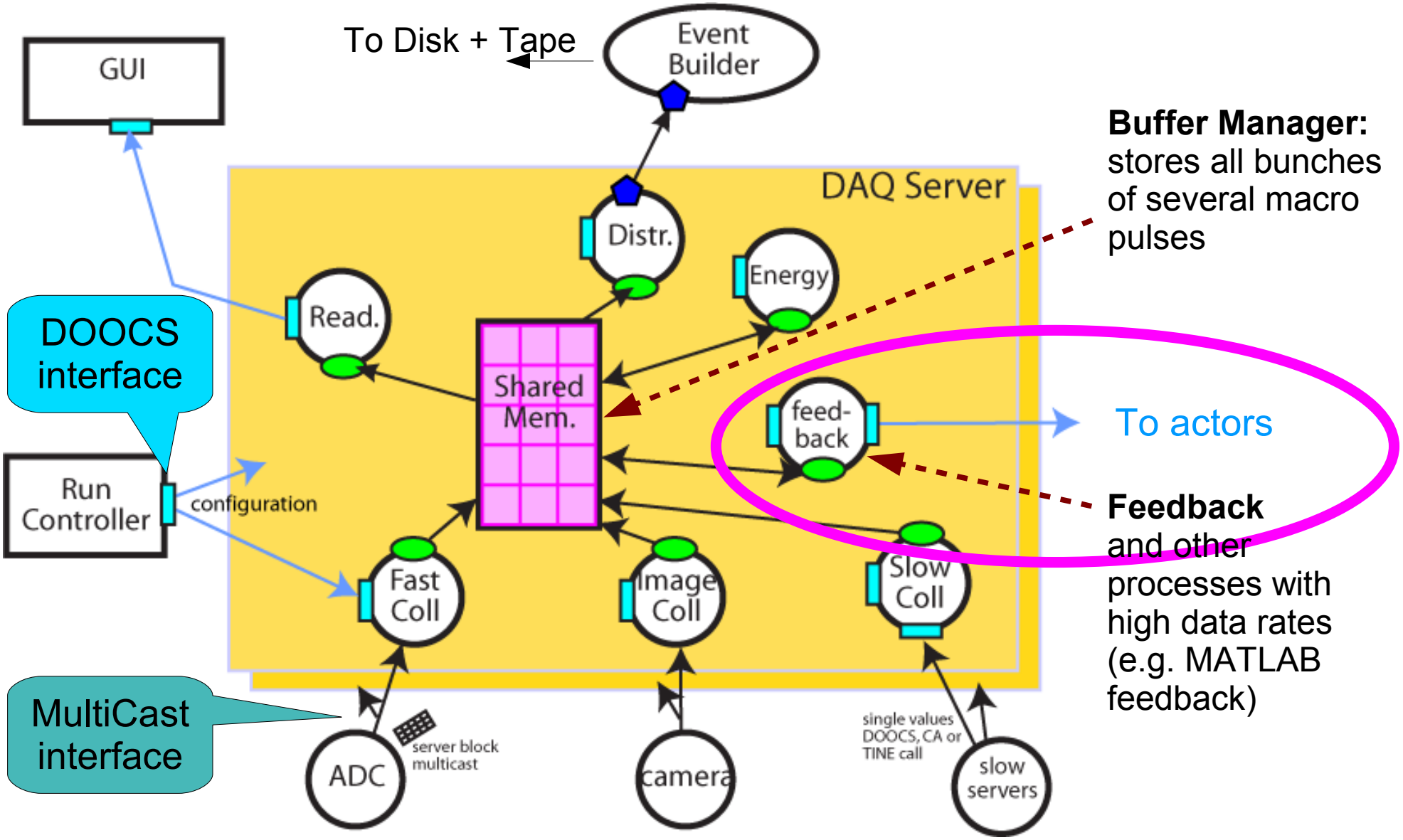
# Compile and Run



# DAQ Overview



# Slow Feedback Integration




**Buffer Manager:** stores all bunches of several macro pulses

**Feedback** and other processes with high data rates (e.g. MATLAB feedback)

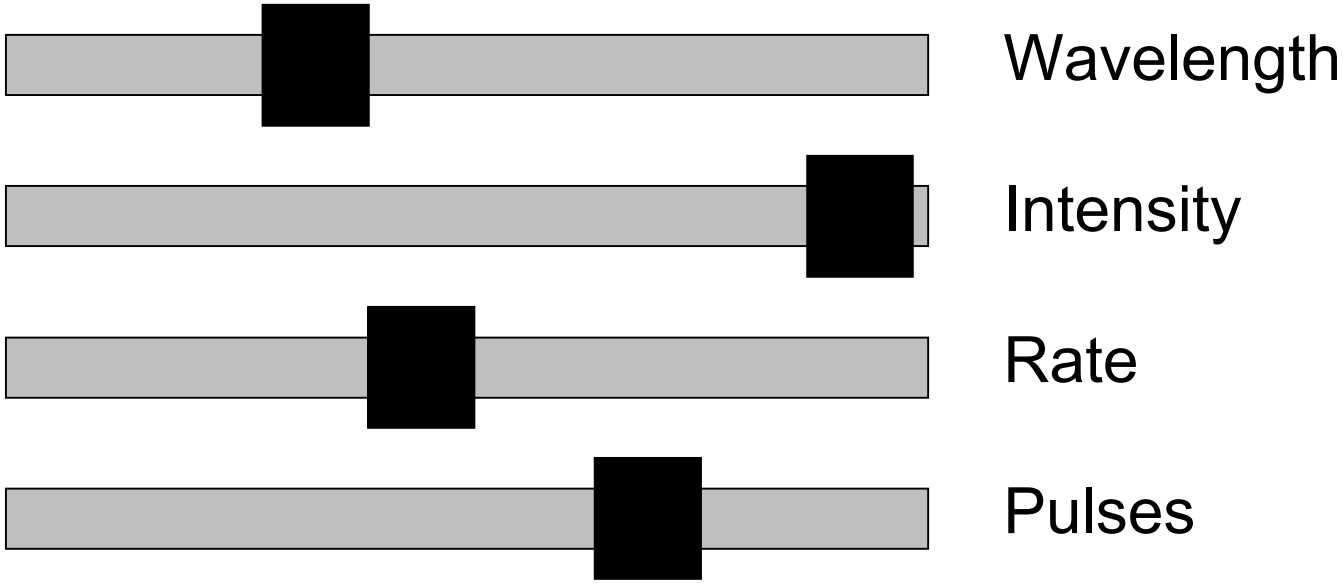
# From Operations‘ Point of View

Markus Hüning

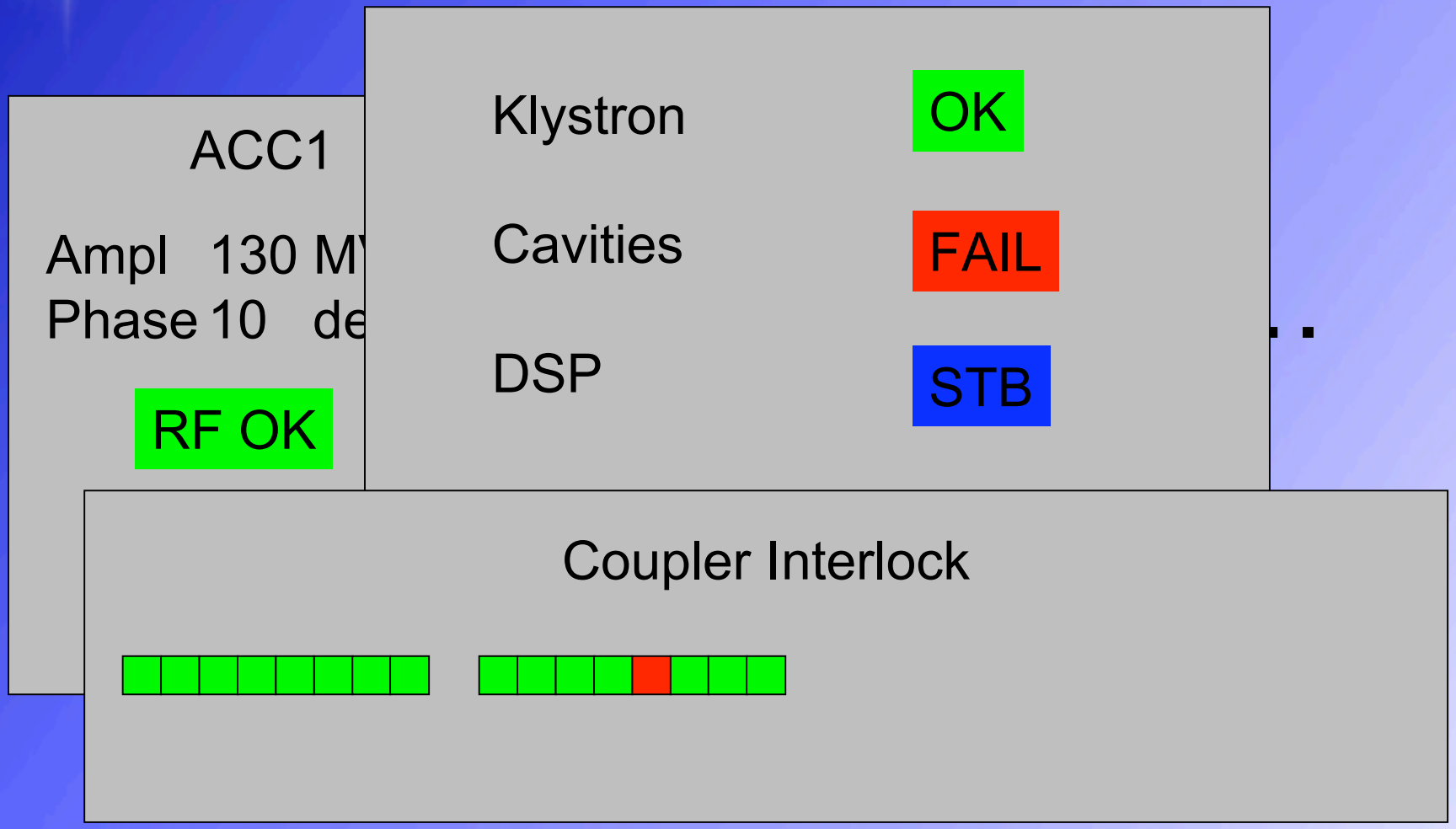
# Who is involved?

- User: FEL Experimentalist, interested in photon beam
- Routine Operation: General Functionality of Machine Subsystems
- Experienced Operator: Looking for State of Modules
- Expert: Debugging

# User



# Experienced Operator



# How to get there

- Don‘t disturb operation
  - minimize dedicated beamtime
  - avoid irreversible interruptions
  - minimize bugs
- Provide early benefit
- Earn trust of operators (and coordinator)
  - functionality visible (and comprehensible) for non-experts
  - more controls than just on/off
  - robust system
  - be quicker than the operator
  - be flexible responding to his wishes

# Experience with TTF1 Automation and FLASH Operation

Automation Mini-Workshop, DESY, May. 22-24, 2006

Valeri Ayvazyan
&
LLRF Team

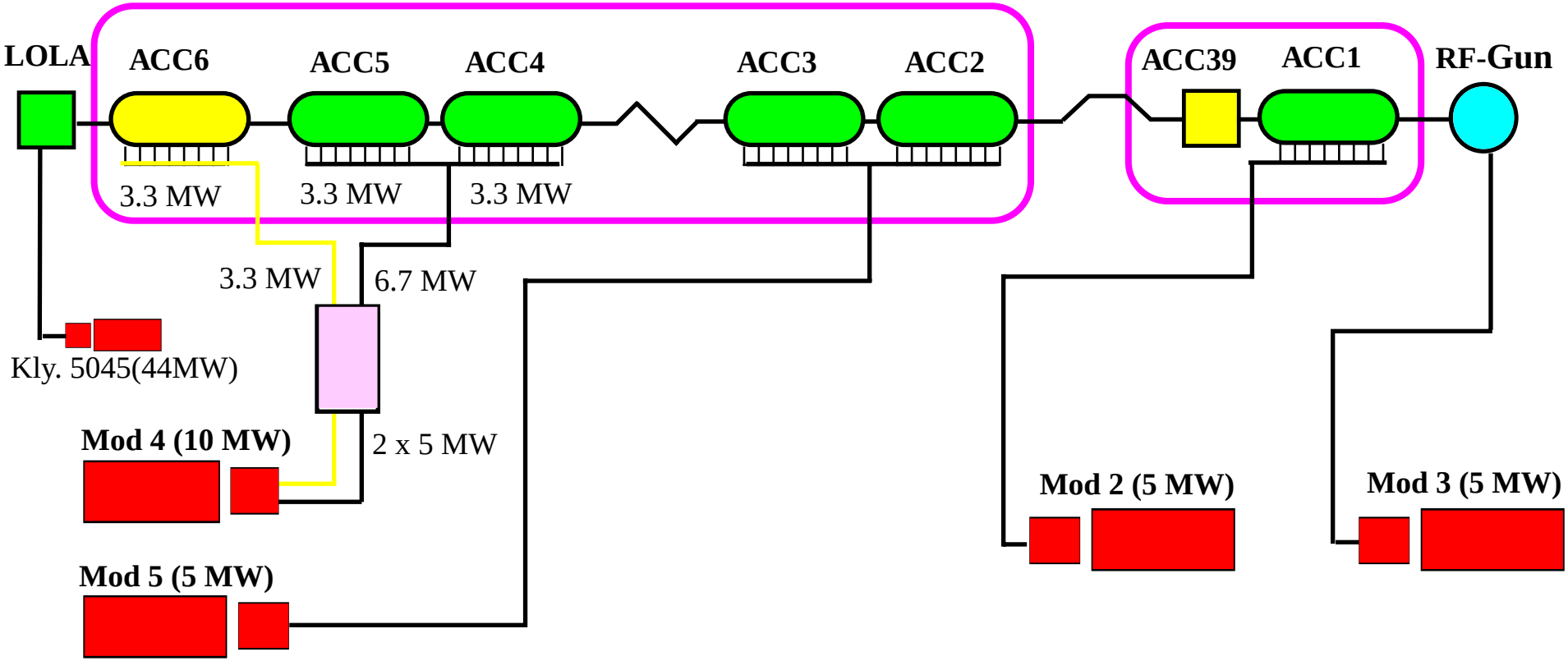
**Outline**

- Operational Requirements
- Review of TTF1 DOOCS FSM Servers
- Challenges of RF Control
- Experience with Operators Acceptance
- Conclusion

# Basic Operational Requirements

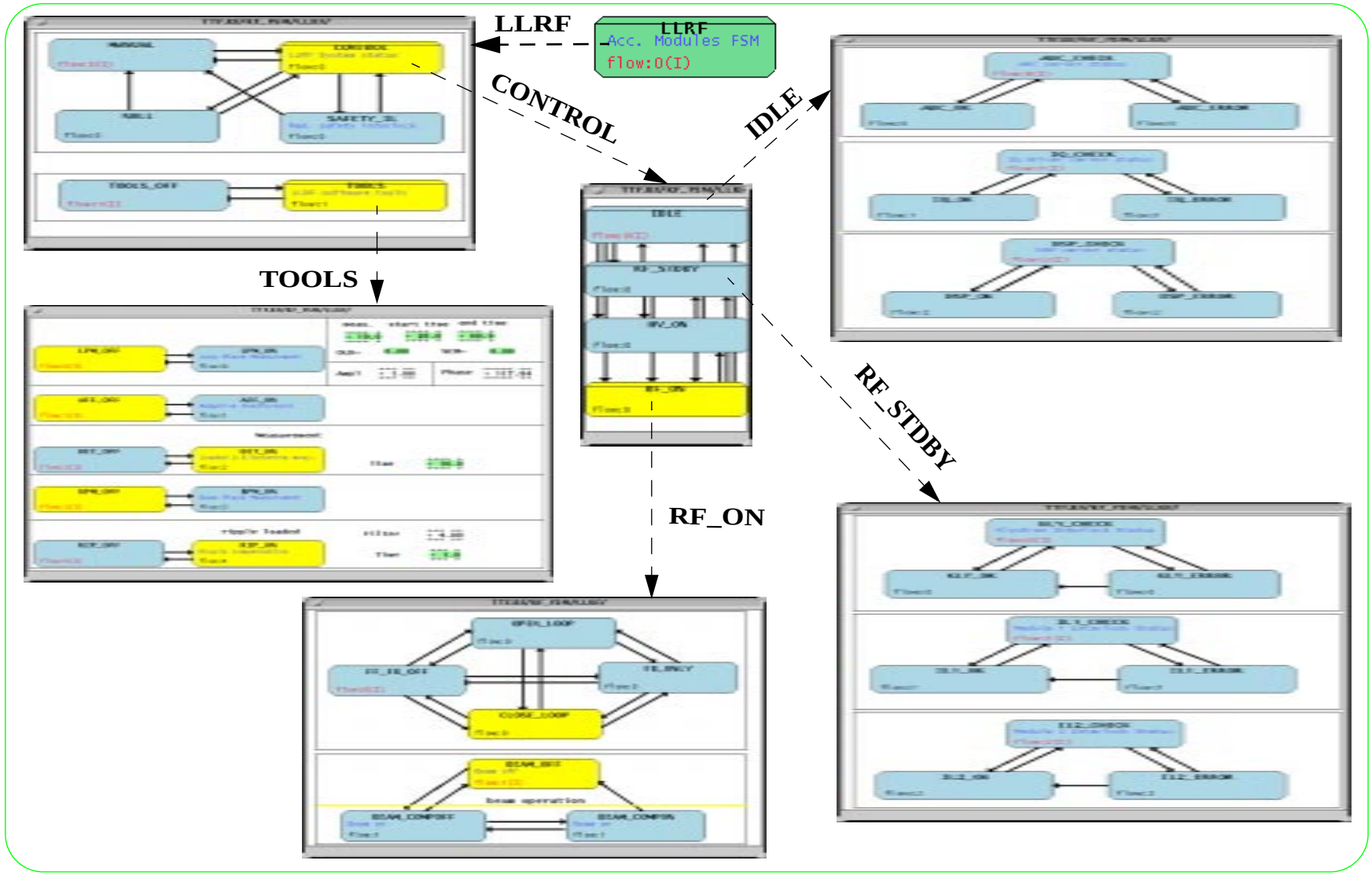
- Reduce workload of operators
- Maximize availability of accelerator
- Increase up-time of the accelerator
- Improve reproducibility of high quality beam after interrupts
- Predict possible faults
- Improve Machine Performance
- Increase Reliability

# FLASH RF System Diagram




- FSM for Cryogenics Operation
- Interaction RF & Cryogenic Systems

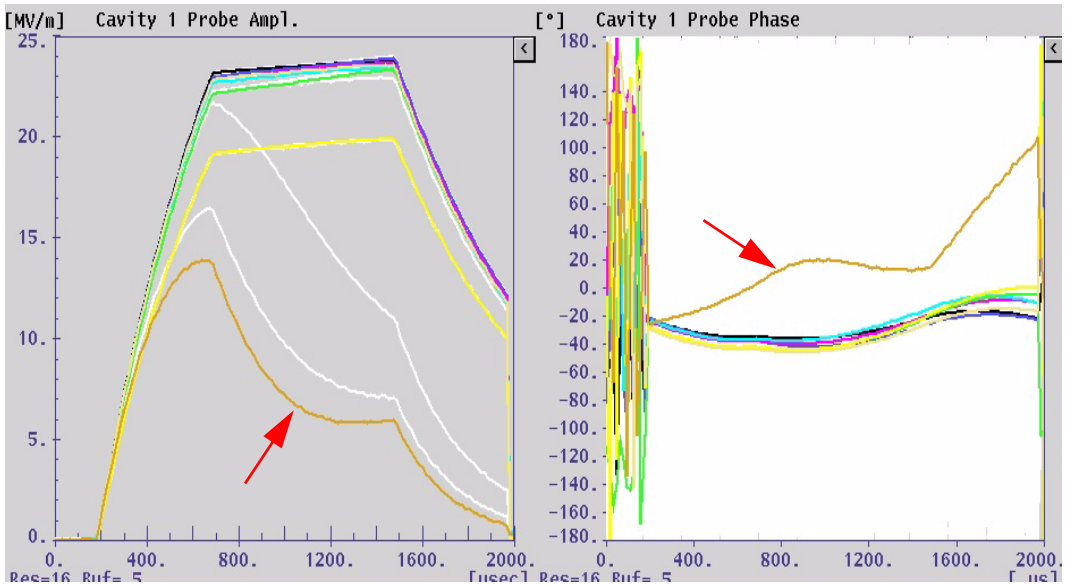
# TTF1 RF FSM Server



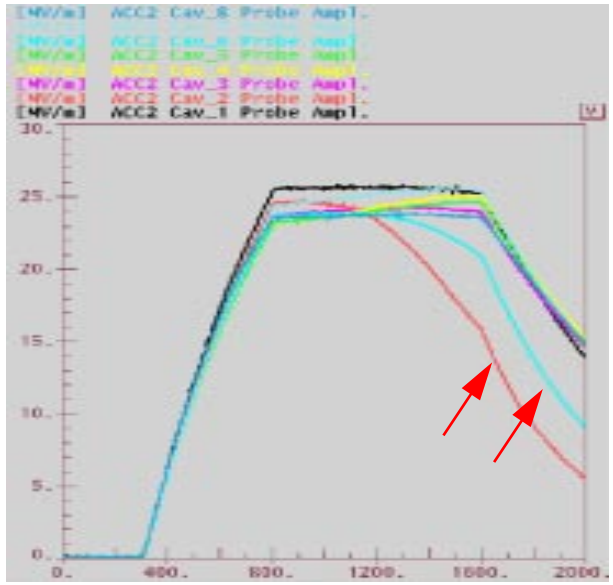
# Exception Handling

- Turn-off klystron if cavity/coupler fault occurs (intra-pulse and inter-pulse)
- Reduce gradient in rf station where fault occurs
- Reduce pulse length (rf and beam) if fault occurs
- Detune cavity in which fault occurs
- Change loaded Q of cavity in which fault occurs

5Hz

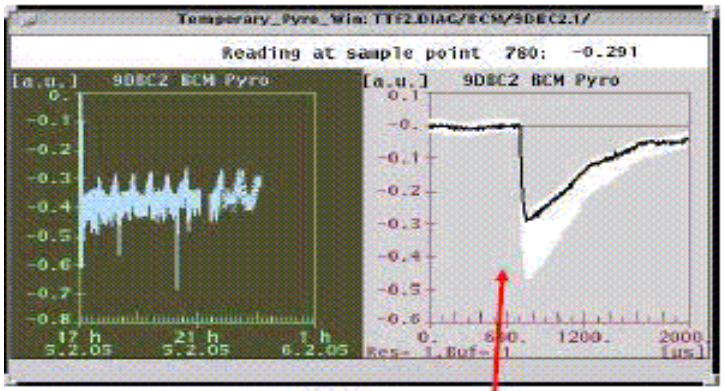


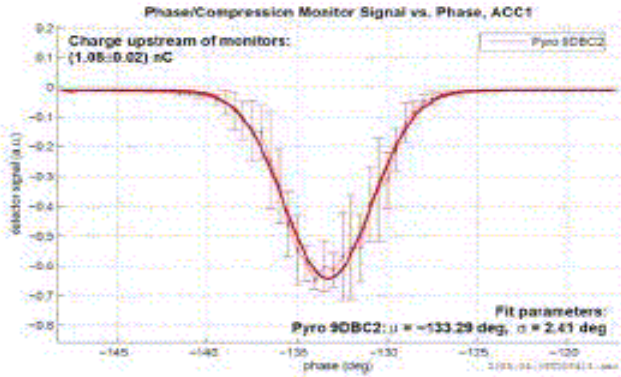

2Hz

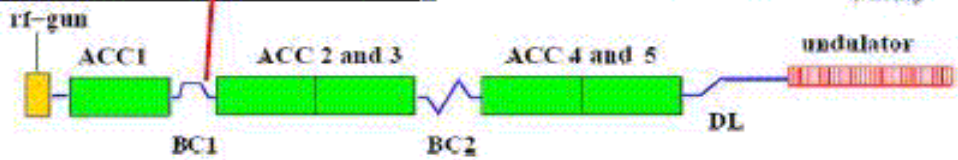


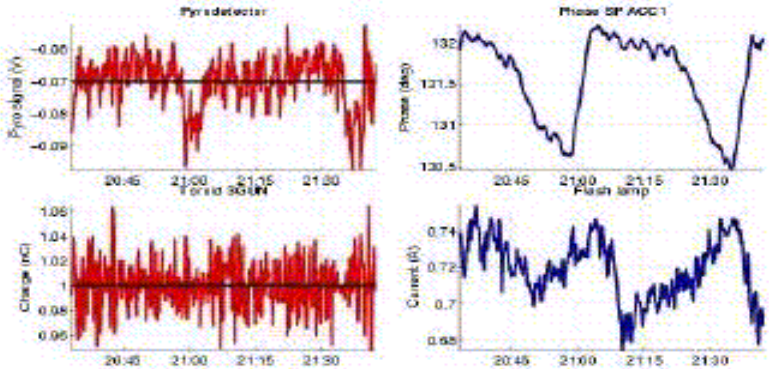
# Control Performance Issues

- Phase Stability
- Slow Feedbacks
- Loop Phase Correction



# Conclusion

- Distributed FSM

- Hardware / Software (DSP/FPGA,Front-End,Main Frame)

- Configuring FSM

  - Different Levels/Options

- Mixed Modes of Operations (Manual/Automatic)

- Independent from FSM Procedures

- Integration with DAQ System

- More Diagnostics are Required

# Automation in an Evolving Environment

alexander.brandt@desy.de

# Airbus

- An accelerator is not an Airbus.
  - An Airbus is a (hopefully) mature device that is manufactured several hundred times. It is delivered to users once it is considered “ready”.
- In an accelerator, there is a high coupling between users and developers

# What should be Part of the Automation?

- Things that happen once (e.g. on install)?
  - Cavity conditioning: done by the expert by simple scripting?
- Things that happen a few times per year?
  - Tuning of cavites - scripting?
  - Vectorsum calibration - scripting?
- Things that happen a few times per month
  - Cavity quenching - button?
- Things that happen a few times per week, per day, …
- Things we don't know yet?

# Automation of the Automation Environment

- **Automation of something in an "evolving environment" means also "automation of the *development environment*".**
  - **Changing something in the automation has to go very fast, on-the-fly.**
- **Paradoxon: the harder one tries to keep it flexible, the faster it will converge towards something stable**

# The Framework is the Key

- The existing DOOCS-FSM framework is nice but
  - It requires too much DOOCS knowledge
  - Small changes require recompilation :(
  - It cannot follow the improvements as fast as they occur (bugfixes, new elements)
- LLRF-FSM-Proposal: 3-Layer Framework (Procedures/Procedure Server/FSM Logic)

# Summary

- The key is to have
  - a flexible framework that is maintained by the DOOCS-people :)
  - smart people that use the framework (during there shifts?) in order to automate things

# AI support for automation and automation design

Research in AI is concerned with producing machines to automate tasks requiring “intelligent” behaviour. Examples include control, planning and scheduling, the ability to answer diagnostic and consumer questions.

**Intelligent behaviour**

- ability to reason - reason about the state.
- - || - to plan - sequence planning.
- - || - to solve problems - exptn handling / autorecovery.
- - || - to learn - not relevant.
- - || - to “think” abstractly - ???/reason with abstractions.

# HOW DO PEOPLE DEAL WITH COMPLEXITY

- BLACKBOXING (making subsystems)
- INTERFACING (common interfaces)
- DEVISING NEW ABSTRACTIONS AND USING DIFFERENT LANGUAGES (Dromey tree)

## **HW CONDITION ESTIMATION** (WORLD/DATA DRIVEN)

- What is the element of abstraction (condition of the system).
- How to find set of (elements of abstraction) relevant to system functionality.
- How to express each relevant state in a terms of system observables – strict semantics.

## **SEQUENCE PLANNING** (GOAL BASED)

- Finding a sequence of procedures taking HW from current state to the target state.
- Estimation whether is feasible.

## **EXCEPTION HANDLING** (PATTERN MATCHING)

- Howto recover the FSM from the states which are not necessarily relevant to the system functionality (automaton complexity reduction).

## NATURALLY SUPPORTS:

- Goal based reasoning - backward chaining.
- Data based reasoning - forward chaining.
- Pattern description - rules for pattern descritption.

## ALLOWS EASY DECOMPOSITION BETWEEN:

- FRAMEWORK PART – model of computation (alternative approaches – simple FSM, Alex's flowchart, model with planning, Dromey's tree).
- EXPERT KNOWLEDGE – structures to be digested by model of computation.

**HOMOGENOUS STRUCTURES**

## METAPROGRAMMING:

- Syntax check – before run.
- Semantics check – completness - cohestion.
- Testing support.

**DON'T PAY FOR WHAT YOU DON'T USE**

**DO WE NEED TO KNOW CONDITION OF THE STATE ? CHOICE BETWEEN FLOWCHART AND STATE/TRANSITION SYSTEM.**

**ABSTRACTIONS AND RELATIONSHIPS:**

- Procedure definition – preconditions / postconditions /meaning/usage – when/how often/by who. **STEFANS LIST.**
- System state definition
  definition based on observables (define domains).
- State-space estimation (relevant to the problem and minimal).
- Exception – specification.
  (answear the question – how expert recognizes and solves the most frequent exceptions)

**DERIVED FROM PRE-POST CONDITIONS OF THE PROCEDURES + FEW EXTRA (ERROR/RECOVERY /MANUAL)**

**DO WE NEED TO KNOW CONDITION OF THE STATE ? CHOICE BETWEEN FLOWCHART AND STATE/TRANSITION SYSTEM.**

- HW CONDITION ESTIMATION
- SEQUENCE PLANNING/DECIDING ON SOME STATIC SEQUENCE.
- Exception classification, processing.

# 1st LLRF Automation Workshop, May 22nd-24th
# S u m m a r y

alexander.brandt@desy.de

# Goals of the Workshop

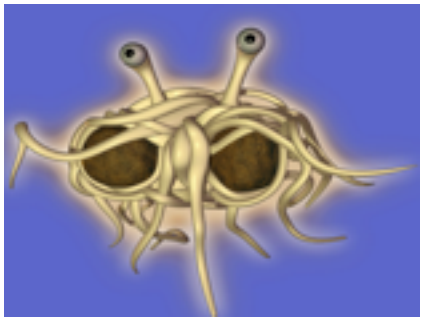
- Define meaning of automation
- Define requirements of LLRF and framework for implementation
- Define what needs to be automated
- Learn from existing experimental versions
- Identify what is missing and what we don't know
- Develop possible architecture(s) for automation
- Develop strategic plan for automation development
- Agree on common framework for implementation of automation

# Requirements

- **LLRF System is one of the most complicated system in an accelerator**
- **Lack of written specification**

- Reduce workload of operators
- Maximize availability of accelerator
- Increase up-time of the accelerator
- Improve reproducibility of high quality beam after interrupts
- Predict possible faults
- Improve Machine Performance
- Increase Reliability

- Field Calibration
- Beam Phase Measurement
- Beam Loading Compensation
- Exception Handling
- Cavity Frequency Tuning
  - motor tuner
  - piezo tuner
- Waveguide Tuner Control
- Adaptive Feed forward (Fast/Slow)

# Next steps…

- Document (some) LLRF ideas/procedures
- Build framework task-force
- Involve other groups (diagnostics, laser)
- Looking at the objectives:
  - Develop possible architecture(s) for automation
  - Develop strategic plan for automation development
  - Agree on common framework for implementation of automation
- …what do you think?