
An FSM-based Framework and Procedures for LLRF Automation

alexander.brandt@desy.de



Overview

- Requirements for LLRF-automation
- ~~Detailed FSM theory~~
A handy FSM example
- LLRF-automation: separation of tasks
 - Procedure Framework
 - DOOCS FSM Implementation
- Procedures (briefly)
- Outlook

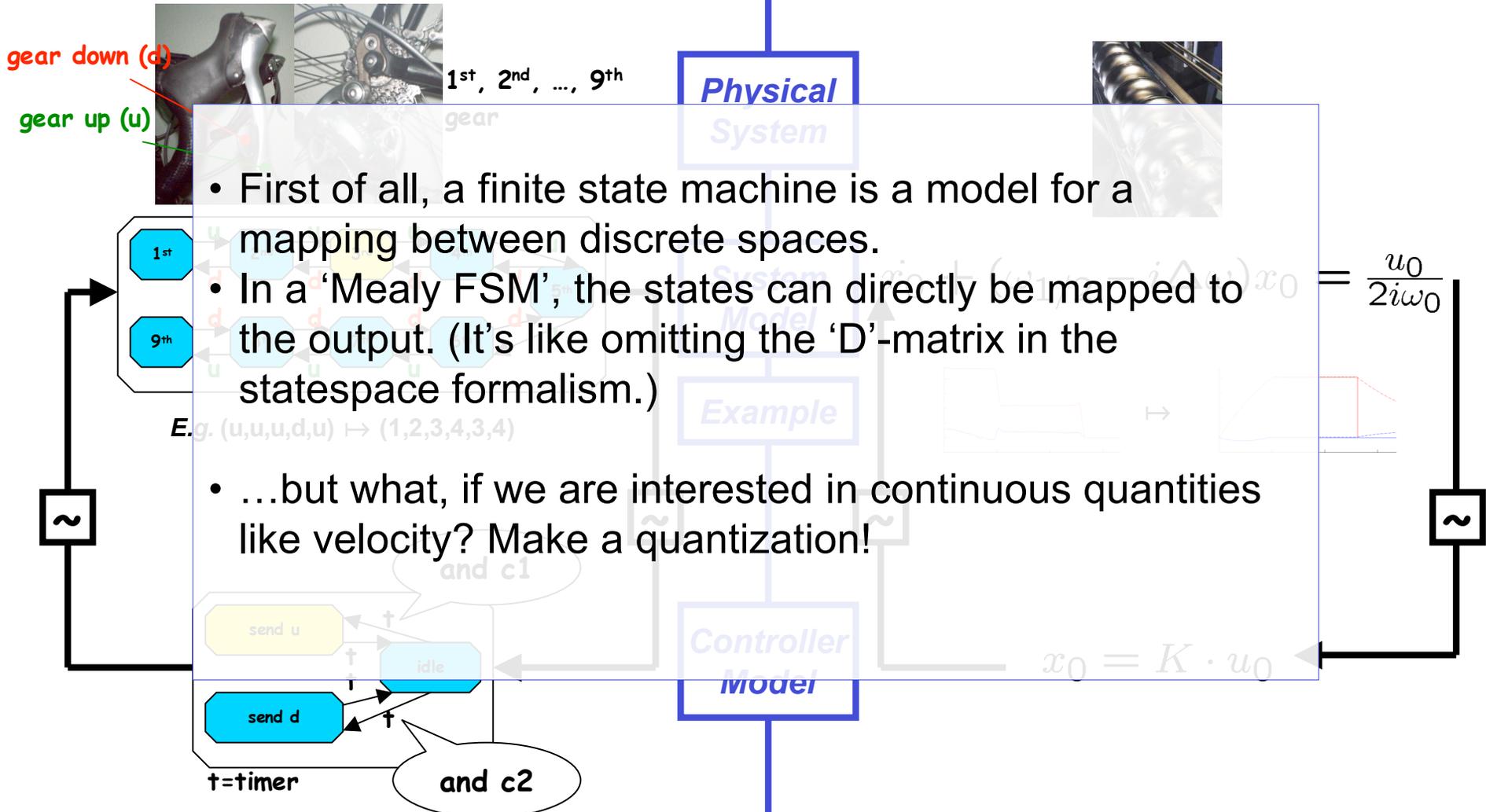


Requirements for Automation

- Why?
 - Devices are getting **digital** now, you need a PC rather than a screwdriver
 - Ease operators job, reduce operation errors
 - “A large number of machine-errors is caused by the operators themselves”
 - Manageability, reproducibility, availability, ..., *ility
- How?
 - Be applicable **on top** of existing infrastructure
 - An a priori consideration of automation is nice but unrealistic due to the large number of subsystems
 - Accomplish operators **acceptance**
 - **Transparency** for subsystem experts
 - **Expandability** and **adaptability** for subsystem experts
 - Deal with several ways of **bypassing** the automation



A Handy FSM Example



Quantization

- What can the quantization be for the LLRF? Most likely procedures with discrete output-values.
- Two elements of the LLRF-Automation already identified:
 1. Procedures (plus infrastructure)
 2. FSM for automation
- The DOOCS control system already provides a nice FSM generator.

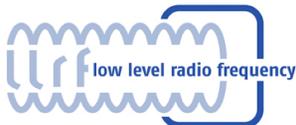
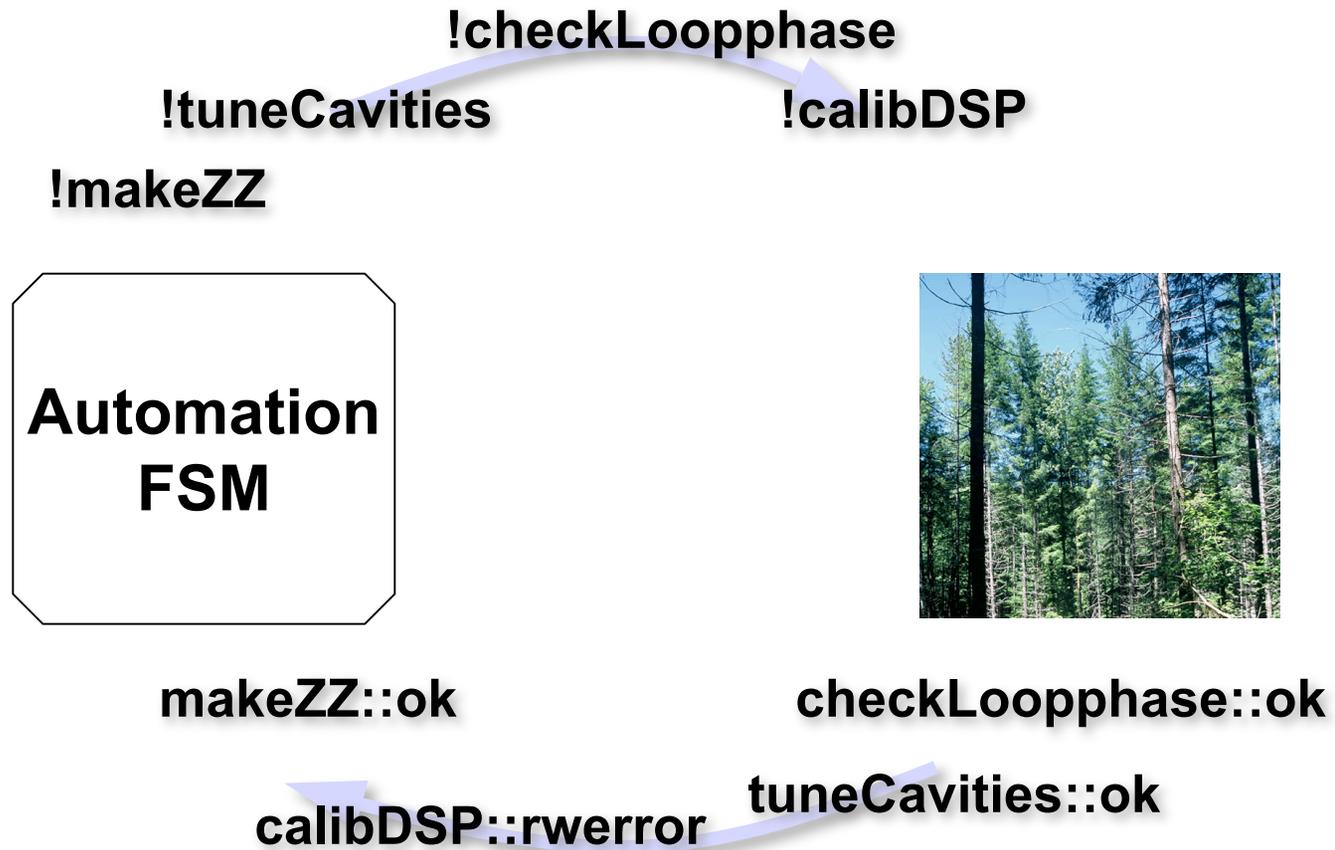
This box can be modelled by an (non-deterministic) FSM!!

**Automation
FSM**

Discrete Dataflow
Continuous Dataflow



Procedure Framework (Ideally)



Procedure Framework (Really)

sat_panel1: TTF2,RF/FSM,ACC1,SAT/GEN_SAT/

Stateless Procedures -Server

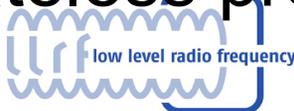
	Configuration	Short Description	Human-Readable Result	Numerical Res.		
FUN0	11rf/runDummy_csh_out0.dat 11rf/out0.dat		operation Successful!	0	-0.02	1.39
FUN1	11rf/runRandomDummy_csh_out1.dat 11rf/out1.dat		operation Successful!	0	0	0
FUN2	11rf/runOFFCa1_csh_out2.dat 11rf/out2.dat	Correcting Offsets.	offset adjustment finished! (Maybe you need to run this one more time? Please	0	0	0
FUN3	11rf/runSetSGandLP_csh_out3.dat 11rf/out3.dat	Correcting loopphase.	Loop Phase and System Gain look fine.	0	-1.97	0.12
FUN4	11rf/runAdaptFF_csh_out4.dat 11rf/out4.dat	Adapting Feedforward (DSP-par.).	Successfully adapted the Feedforward!	10	0	0
FUN5	11rf/runAdaptFF_csh_out5.dat 11rf/out5.dat	Adapting Feedforward	Feedforward Adaption Successful	0	0	0
FUN6	11rf/runDefaultFF_csh_out6.dat 11rf/out6.dat	Applying default Feedforward.	Successfully set default feedforward.	0	0	0
FUN7	11rf/runCouplerKb_csh_out7.dat 1 11rf/out7.dat	Checking coupler interlocks.	No interlock observed.	0	0	0
FUN8	11rf/runSetSGandLP_csh_out8.dat 1 11rf/out8.dat	Checking loopphase.	No drive from 11rf is applied.	-13	0	0
FUN9	11rf/runFieldQualify_csh_out9.dat 11rf/out9.dat	Checking data quality.	FB=0 or FF=0: Data quality is unacceptable.	9	inf	105.87
FUN10	11rf/runCheckNetwork_csh_out10.dat 11rf/out10.dat	checking network.	Network works fine.	0	0	0
FUN11	11rf/runRampFB_csh_out11.dat 11rf/out11.dat	Ramping Feedback.	No permission to touch FB!	1	0	0
FUN12	11rf/runSetSPw1_csh_out12.dat 11rf/out12.dat	Setting amplitude and phase.	Some values allowed to touch SP-values!	1	0	0
FUN13	11rf/runOperatorAction_csh_out13.dat 11rf/out13.dat	Looking for operator action.	No operator action detected.	0	0	0
FUN14	11rf/runKathLaser_csh_out14.dat 11rf/out14.dat	Checking cathode laser settings.	Cathode laser in range.	0	0	0
FUN15	11rf/runGetDSPcalData_csh_out15.dat 11rf/out15.dat	Retrieving DSP calibration-data.	Saved data in 22-Nov-2005T1055-dspcaldata-ACC1.mat.	0	0	0
FUN16	11rf/runEvalDSPcalData?_csh_out16.dat 11rf/out16.dat	Evaluating DSP-Cal-Data.	Finished data evaluation.	0	0	0
FUN17	11rf/runSignalCalib?_csh_out17.dat 11rf/out17.dat	Calculating cavity signal calibration.	stored data in cavcalibdata-ACC1-8.mat	0	0	0
FUN18				0	0	0
FUN19				0	0	0
FUN20				0	0	0
FUN21				0	0	0
FUN22	offCa1_ACC456_out22.dat out22.dat		Script execution error.	-99	0	0
FUN23	setSGandLP_ACC456_out23.dat out23.dat		calculated bandwidth out of range. (Maybe there is no drive signal at all?)	-11	0	0
FUN24	oneStepAFFDSP_ACC456_out24.dat out24.dat		Calculated bandwidth out of range	-11	0	0
FUN25	oneStepAFF_ACC456_out25.dat out25.dat		Script execution error.	-99	0	0
FUN26	defaultFF_ACC456_out26.dat out26.dat		Script execution error.	-99	0	0
FUN27				0	0	0
FUN28				0	0	0

Annotations:

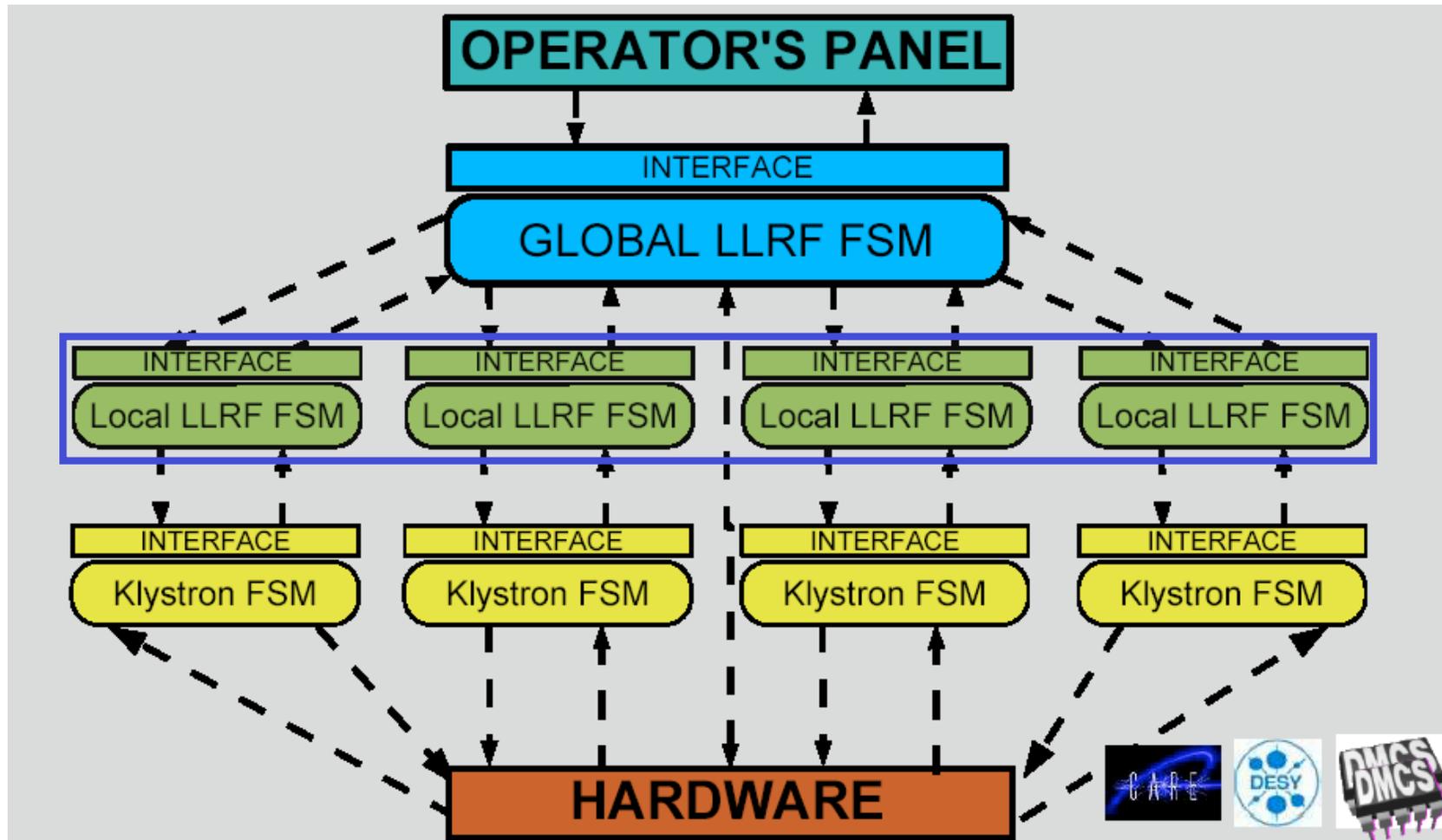
- button
- configuration
- location of binary
- no restrictions
- short description (appears in logfile)
- human readable result (appears in logfile, too)
- discrete result of the procedure
- additional outputs

Procedure Framework

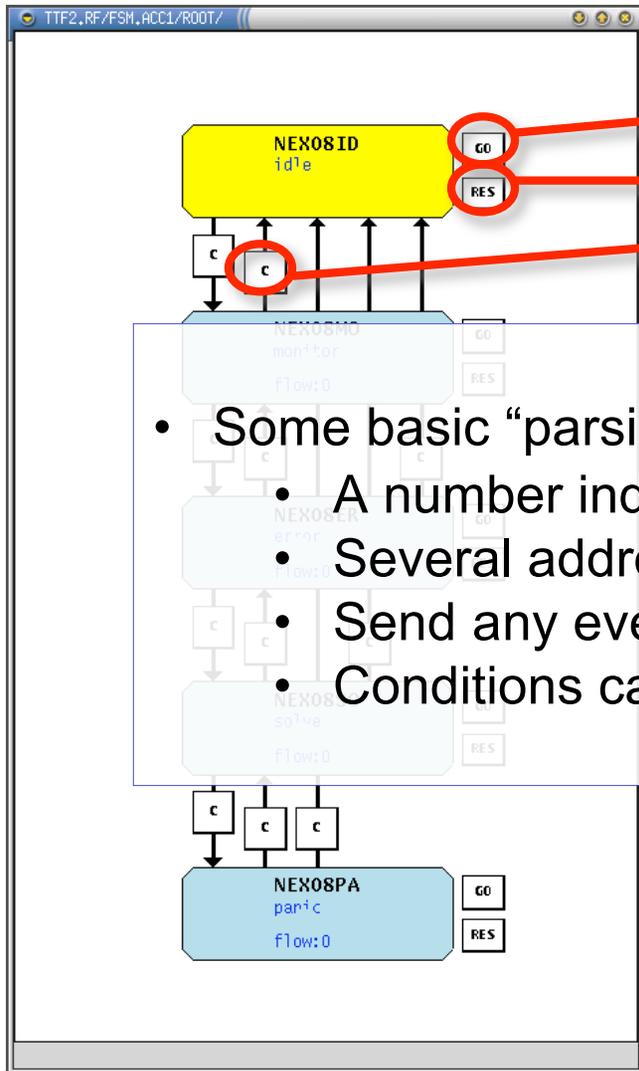
- Would be nice to have addresses like
TTF2.RF/PROCEDURES.ACC1/LOOPPHASE/START
However, I could imagine more sophisticated solutions: every procedure has an identifier. The FSM just “shouts” these identifiers into a channel and gets back results...
- Philosophy of procedures
 - “Stateless” (no interaction while executed, no intrinsic knowledge about previous executions)
 - Independent of each other (but not necessarily)
- Distributed system not yet realized, but would be nice to have
- A framework supported by DOOCS like this would be helpful - not only for FSMs
- I think that application of slow-feedback is a “stateless procedure”



LLRF Automation: Big Picture



LLRF FSM: Basic Idea



which button to push
where to find the result
condition for this transition

- Some basic “parsing” is implemented in TStateLogic
 - A number indicates to wait a few pulses
 - Several addresses means press several buttons
 - Send any event to any state on entering a state
 - Conditions can be ‘else’ or ‘all’ or several numbers

Very small extension to the generated code.

```

fputs("ROOT_EXERDQ:: \"init\" event procedure: ROOT_EXERDQ_init

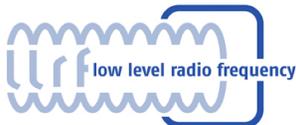
TStateLogic* sd=TStateLogic::factory("EXERDQ");
sd->set_go("ROOT_EXCEPTION.EXERDQ_GO");
sd->set_res("ROOT_EXCEPTION.EXERDQ_RES");
sd->add_event("EXERDQ", "exerdq_exmodq", "ROOT_EXCEPTION.EXERDQ_
sd->add_event("EXERDQ", "exerdq_exsodq", "ROOT_EXCEPTION.EXERDQ_

return ST_OK;
}

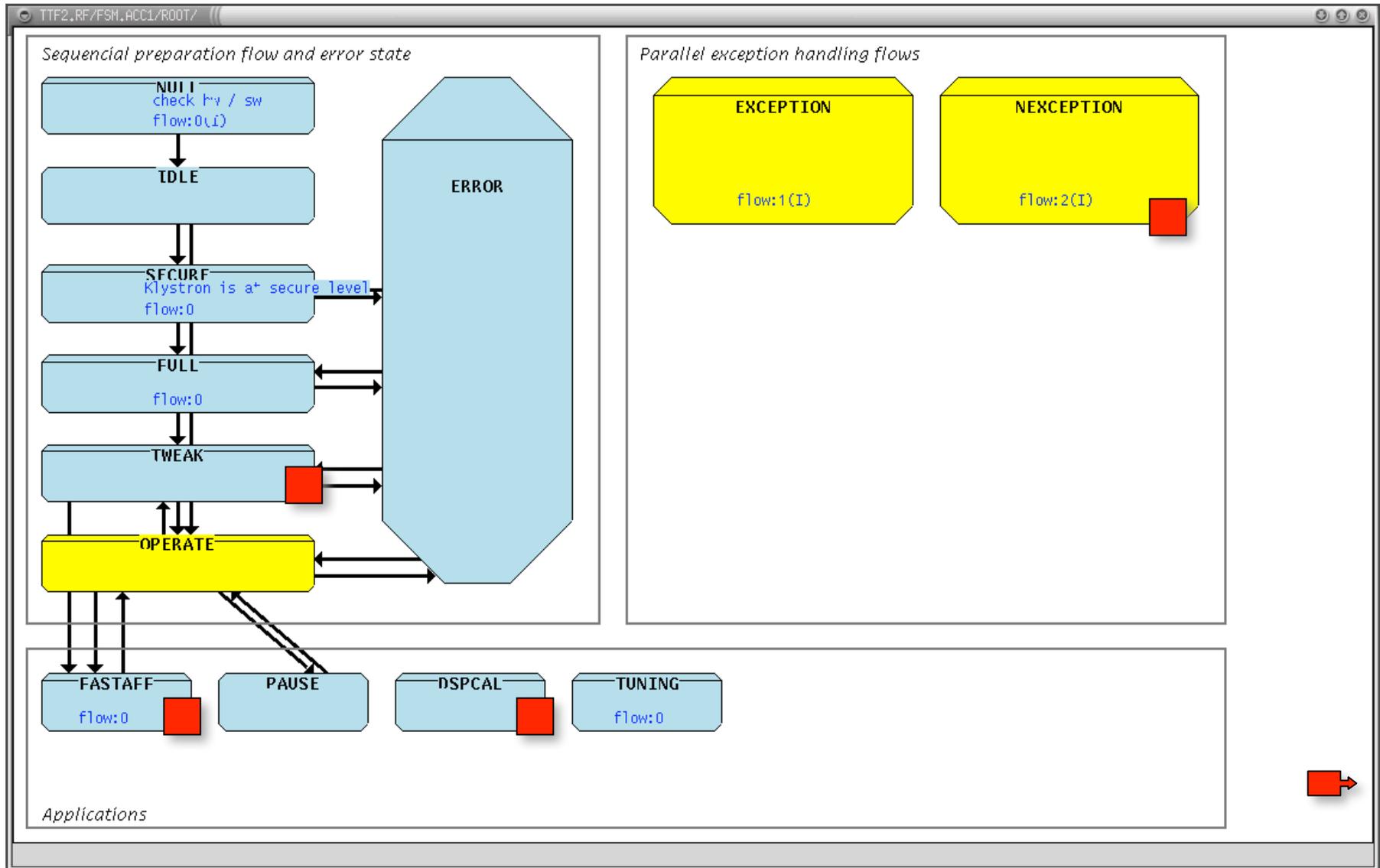
//// DO NOT EDIT THIS AND NEXT TWO ROWS
// enter function for state 'EXERDQ'
int ROOT_EXERDQ_enter_action(FSMstate *st_p)
{

fputs("ROOT_EXERDQ:: \"enter\" event procedure: ROOT_EXERDQ_ent

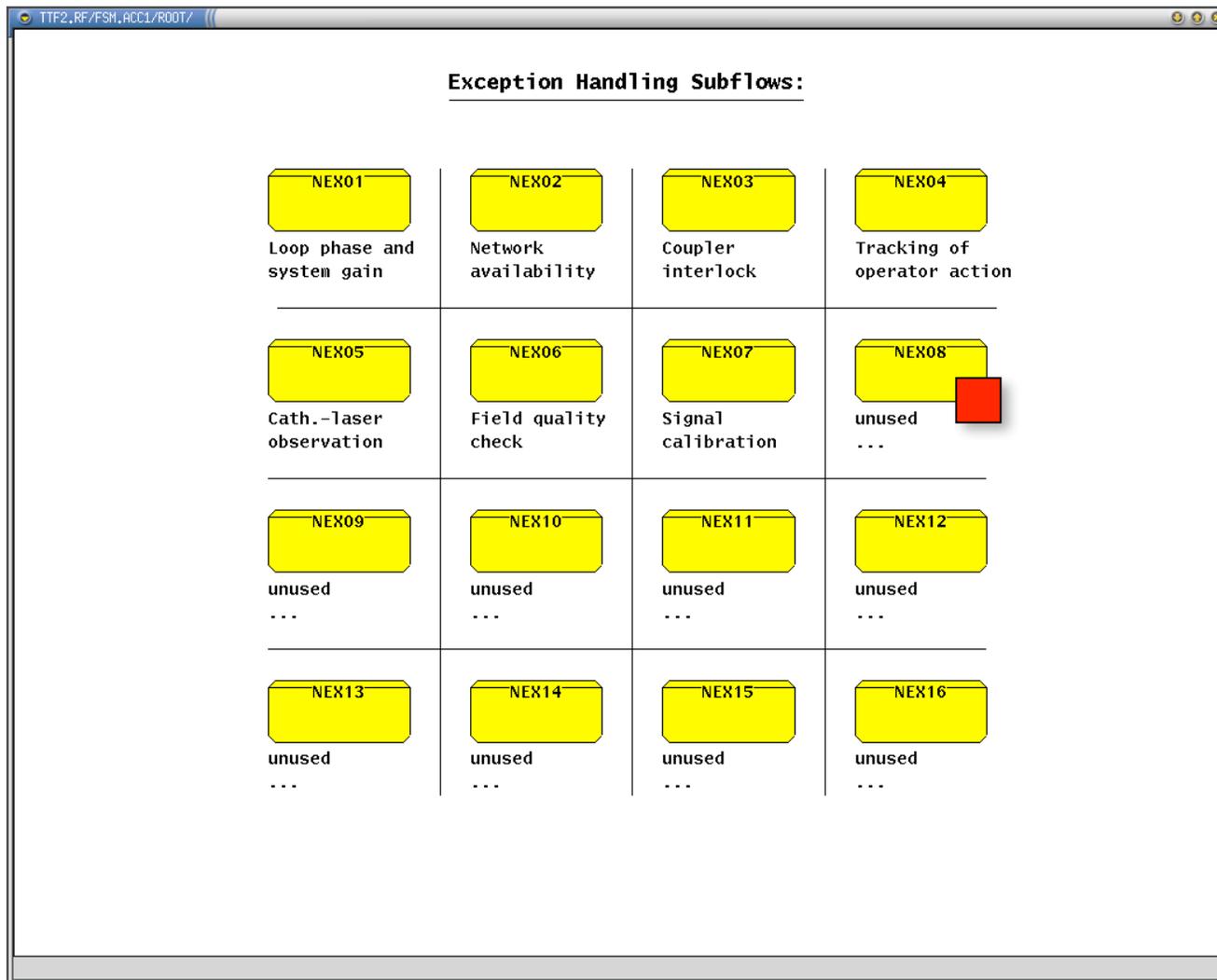
TStateLogic* sd=TStateLogic::factory("EXERDQ");
sd->enter(st_p);
    
```



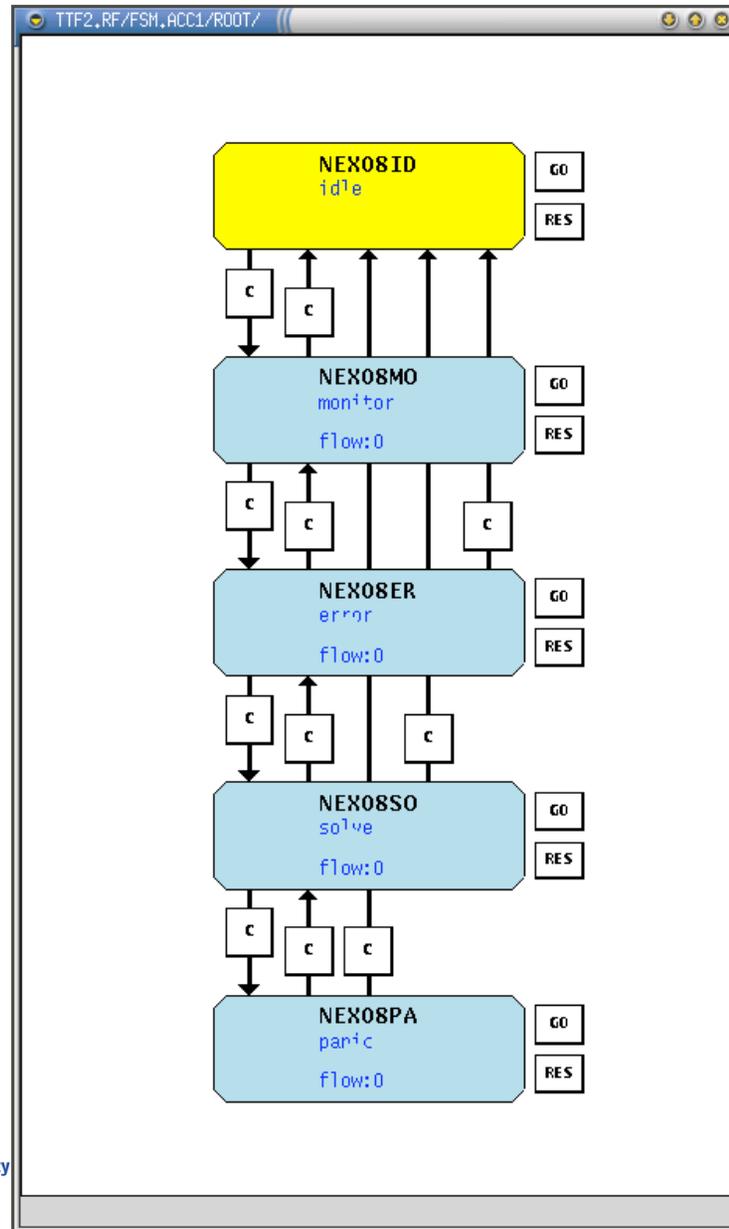
LLRF FSM: Top Level View



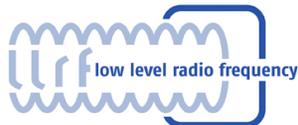
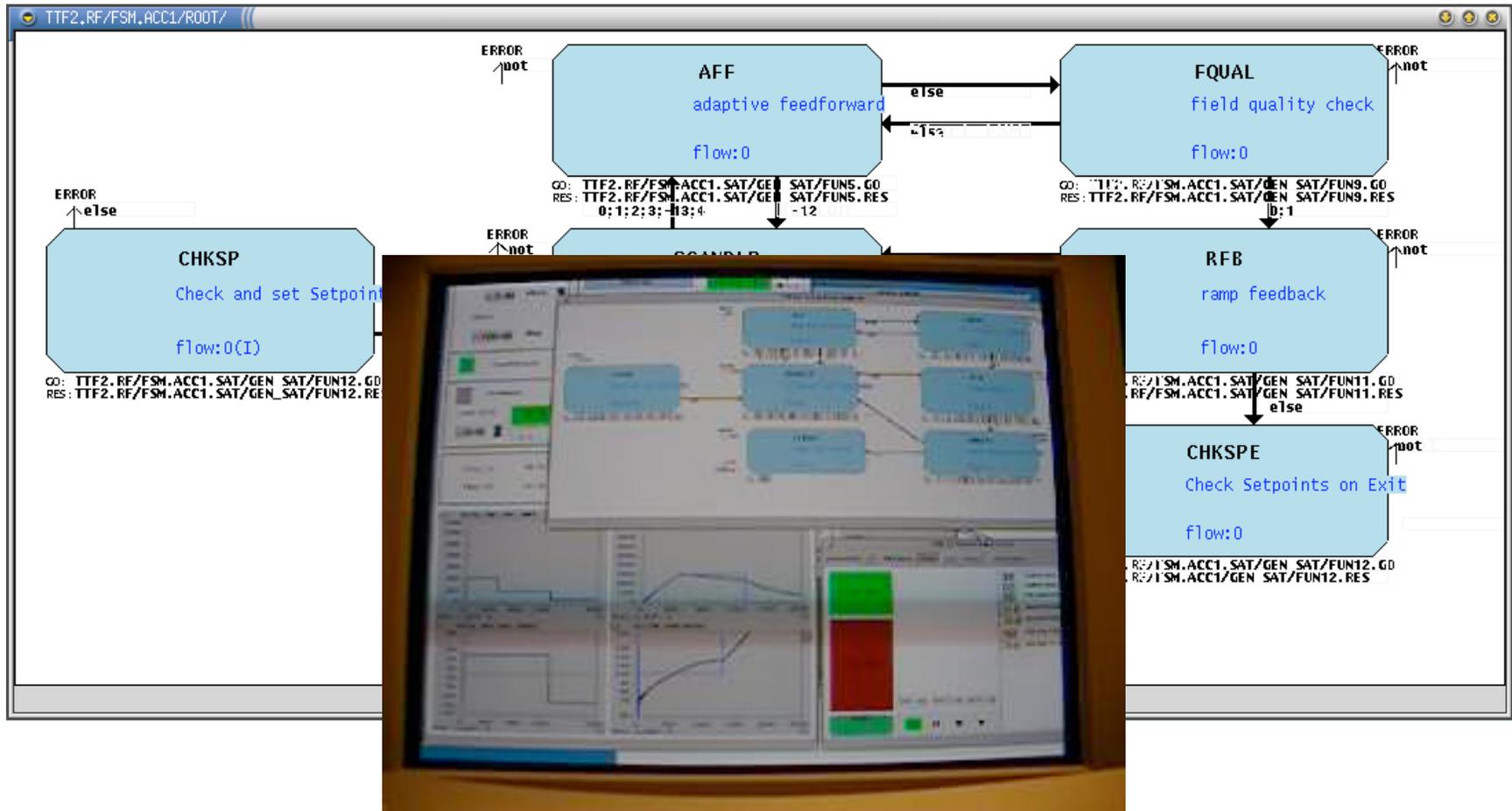
LLRF FSM: Exception Top



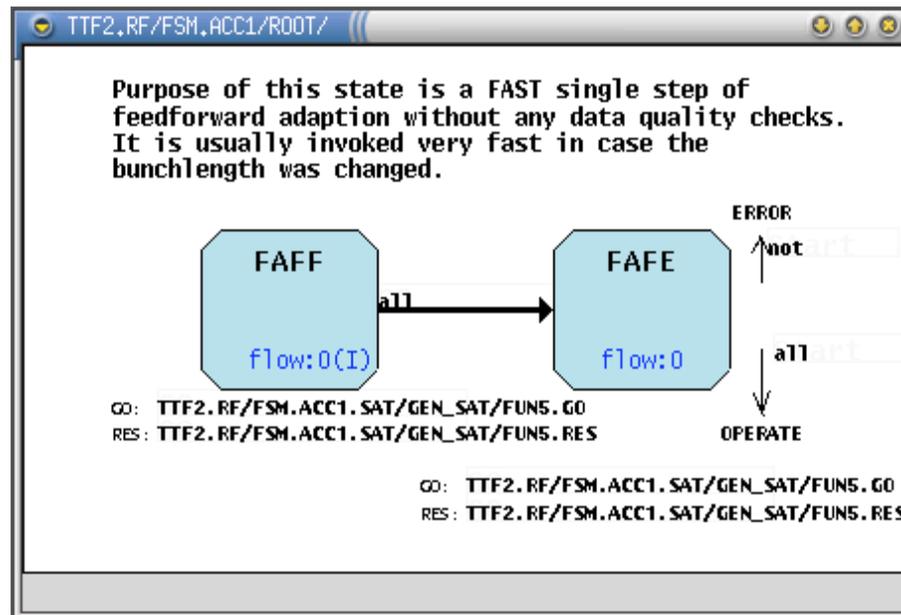
LLRF FSM: Exception Handler



LLRF FSM: Tweak



LLRF FSM: Fast Adaptive Feedforward



low level radio frequency

LLRF FSM: Summary

- Basic philosophy: model the *operator*.
- Goal was not to find the final solution but to build something that can easily be changed
 - Changing procedures, wait-cycles, conditions can be done *on the fly* from any ddd-panel (restricted access might be a good idea)
 - Adding states, changing the structure needs **regeneration** of code and **recompilation** - this is automatic except from the `TStateLogic`-statements
- Not only **changing** but also **exchanging** is possible
 - Boguslaw is working on a more sophisticated solution that omits the DOOCS FSM code generator - why not?

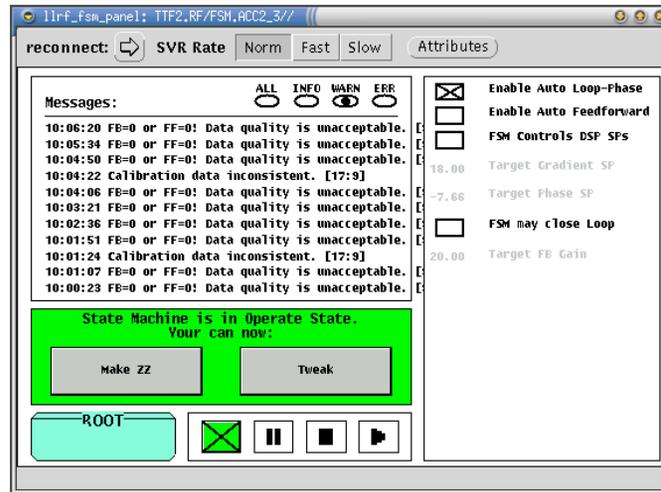


Procedures: Overview

- **Observation Algorithms for settings of DSP and cathode laser**
 - Send events in case of changes
 - Internal tracking allows to distinguish between “operator-intervention” and “automation-intervention” (!!!)
- **Adaptive Feedforward: tested successfully with moderate beam-load**
 - Plans to test algorithm at SNS with heavy beam load in January 2006
- **Loop-phase and system-gain estimation**
 - Running since several month, no operator-complaints so far
- **Algorithms for DSP-calibration**
 - Separated into tasks (data acquisition and evaluation)
 - Script-version has been tested, compiled version needs to be verified
- **Coupler interlock reset**
 - Reset limits per day / per hour / per minute
- **Offset calibration, setting of setpoints, ramping feedback,**
- **What can YOU contribute for YOUR subsystem...?**



Outlook



- FSM-approach meets requirements of transparency and flexibility.
- Procedure-framework has proven to be something useful - I already met people who are interested to use it.
- As next step, I would like to establish the adaptive feedforward in the VUV-FEL. But this needs some time.
- Not yet addressed, but important: Database
- Experience gained for XFEL and ILC automation.

