

BI Software Changes During LS2

BI Technical Board

Outline

General

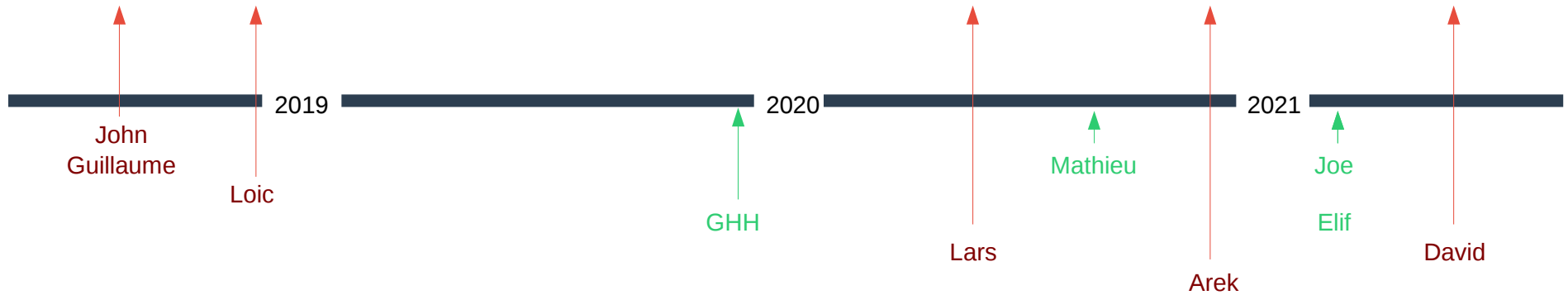
FESA

GUIs

Tools

Future

Manpower Shuffling During LS2



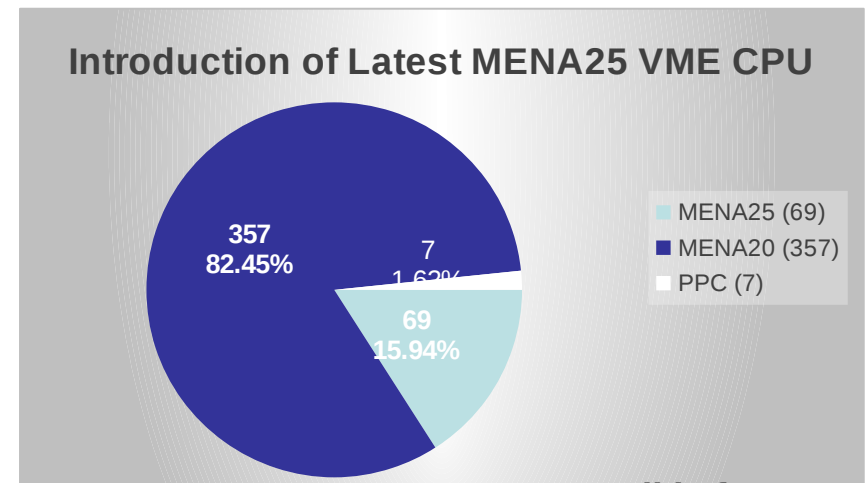
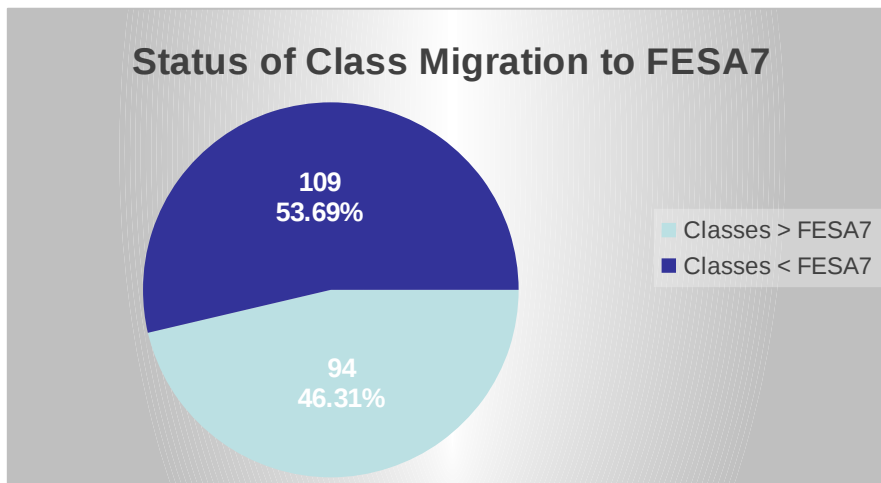
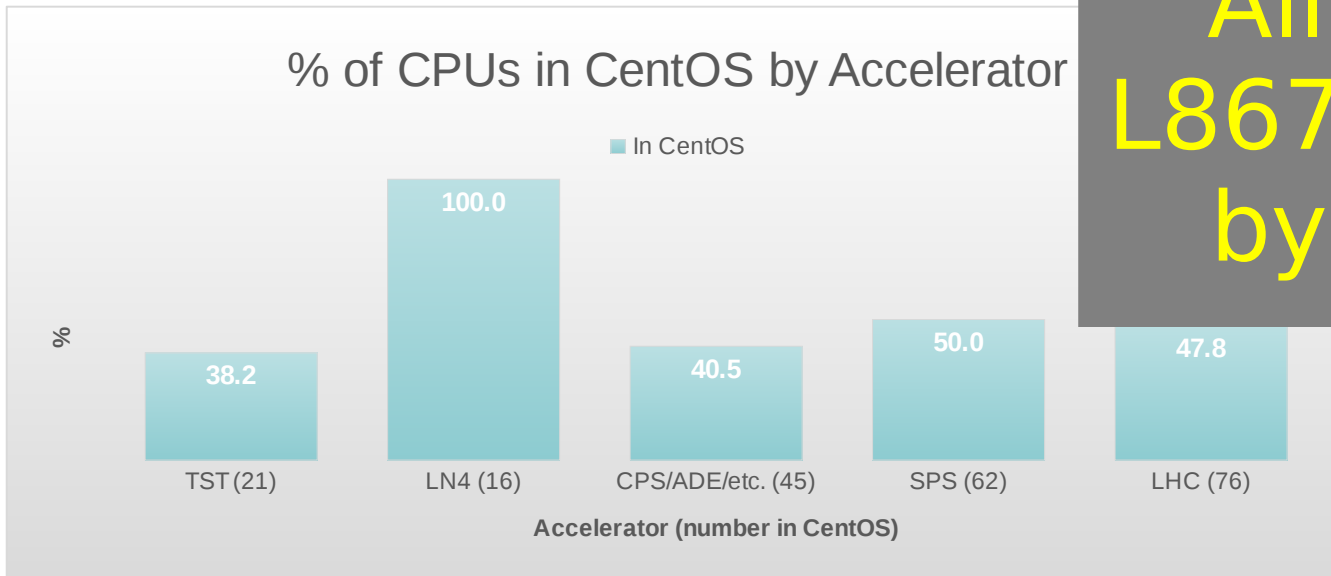
Staff leaving during LS2 caused significant disruption

Challenge for knowledge transfer

Exasperated as have to take on systems blind (not seen with beam!)

Reminder Status End of 2019 (mid LS2)

All systems in L867 and FESA7+ by end of LS2

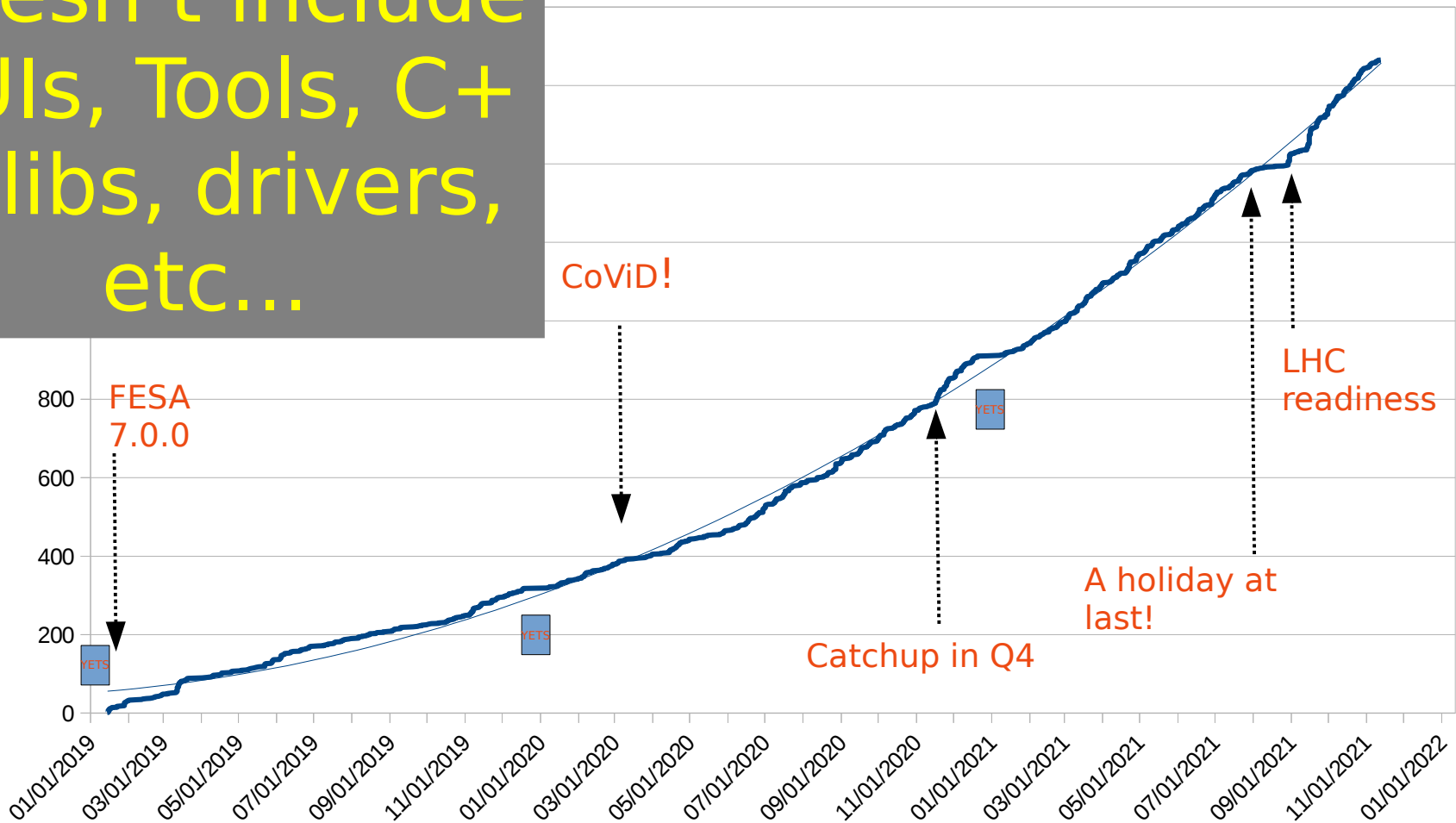


* Slide from BI Day 2019

(Rough) Evolution of FESA Work Done in LS2

Doesn't include GUIs, Tools, C++ libs, drivers, etc...

FESA Commits during LS2



Achieving Results with other Sections

Regular meetings

EA

Wire scanners

Beam Feedbacks

BLM (INJ + LHC + d)

Digital Cameras (early LS2)

Communicating system specifications

Ad-hoc Python scripts as a model from experts (LIU WS)

Full FESA Mock-ups while hardware still being designed (LIU WS)

C code used for defining state machines (BBQ – Miha + Tom)

Pair programming (BLM – David + Mathieu)

General FESA Software Changes

Pre LS2, wide variety of FESA Framework versions used

FESA2, FESA3 (3.X, 4.X, 5.X)

Move to LS2 baseline

64 bit means code changes (bad casting a gogo)

Changes in system behaviour

Some things possible in FESA2, simply not possible in FESA3

Resulted in change of API in some cases

Move to EDGe from Encore

Not compulsory but several new developments or renovations invested in using EDGe

General impression positive - Driver register organisation much-improved

Proliferation of MBLT

Led to changes in FESA design to exploit increased speed

Several changes (bugs) in MENA25 firmware behind our back

Exploiting C++11 (already ancient!)

C++11 available in FESA 8.X

Lambdas and auto

```
auto gt = [](auto a, auto&& b) { return a > b; };  
bool b = gt(4, 2.56); // false
```

Delegation

```
F(int v) : x(v), y(0), p(new char [MAX]) {} // #1 target  
F(): F(0) { cout<<"delegating"<<endl; } // #2 delegating
```

C++11 Std Lib

```
unordered_set, unordered_map, ...
```

New smart pointer classes

```
shared_ptr unique_ptr
```

New algorithms

```
all_of(first, first+n, ispositive());  
any_of(first, first+n, ispositive());  
none_of(first, first+n, ispositive());
```

Can replace many cumbersome lines of code with 'modern' C++11

New FESA features

Simplified export to post-mortem

Now a simple tag in the design exports a property automatically to PM/IQC

No more need for custom code except in special cases

Functions

New in control system

Allows application of settings which change over time

Better than time-aligned arrays

New Controls Features - UCAP

UCAP

Adds another option for system design

Allows removal of some business logic from FESA classes GUIs

Flexible language options (Python, Java)

More usage started to appear towards end of LS2

BSRL

dBLM

PSB Extraction Trajectory

FBCT in SPS/TL

...

Unexpected Problems

Major bugs in early FESA 7.X releases

Bizarre behaviour in 7.0.X where timestamps delayed or corrupted

Lost some time recommissioning systems migrated early in LS2

Discovered many months (years) after release (as needed beam to test)

Export to Post Mortem

Very long timeouts (1 minute) due to hard-coded non-existent PM server in 7.0.X release

Early workaround was to set environment variables or move to FESA 8.X

LUMENS

DAB-based systems had (still have) problems flashing firmware before driver loading

No -ve priority as in transfer.ref

Only solution to have hybrid

LUMENS and transfer.ref!

Other problems still being solved (eg LNA BCT)

Changes to FESA classes

Not possible to list all changes for all classes

All classes needed changes / adjusting

Following slides with 'notable' changes

Doesn't necessarily reflect the amount of time taken

Some classes not in the next slides may have proved troublesome during migration but not 'interesting'

Systems with Notable changes (LINACs)

LN4 SEMGRID

SIS – movement surveillance

New LN4 TxT

For intensity monitoring (@PSB injection point)

New Interlocks defined with OP

LN4 Watchdog

Enhanced with different comparison modes

Adapted for corner cases (injection of 0 turns etc)

Also adapted to PSB and ISOLDE use-cases

BSM

Tuning of BSM + testing of new BSM (BSM2)

Systems with Notable changes (LINACs)

Emittance Meter (test stand)

2 new FESA servers to split the control of the position and the signal acquisition

Exceptionally provided new GUI for OP (not GUI for Experts)

Wireshanners

New devices in LBE

LN4 BPM

Position computation changed to comply with the non linearity requirements

Wireshanners

Support for PSB wire-scanner calibration already during 2019

Systems with Notable changes (PSB)

BEMTxT

LIU turn by turn measurement system

Included also the grid movement to adapt to both PSB and PS

Changes to the electronics for PSB

TMS

Coordination with AlphaData to implement and compile a new L867 TMS library (released end of 2019)

Stable system finally delivered

Worked from day 1 despite important SW and HW mods

dBLM

System taken from (operational!) prototype by Jiri

Needed redesigning from scratch - Python server not aligning with controls 'standard'

OP currently using OASIS - Adaptations needed in firmware - To be followed eventually by FESA class

Systems with Notable changes (CPS)

BGI

Finalised and validated design of FESA class

Software now more 'standard' (Steen + Hampus)

IPBus dependency maintained by CSS

Good also for other classes using IPBus

Systems with Notable changes (SPS)

BLRSPS

6 running sums implemented in SW

(1 fixed cycle-wide and 5 configured wrt threshold/veto/width)

Much improved diagnostics and monitoring

Special tools (PyQt) given to BL for validating VME modules cleanup

Mockup system allowed good testing before handover to OP

Additional LTIM watchdog added

Dump beam if ms are lost on GMT)

BSESPS (servo-spill acquisition based on VD80)

Originally sharing crate with BA2 Diamond BLM

Problems with memory so moved to own crate

Still problems so class was redesigned - All stable now

Systems with Notable changes (SPS)

BPLOFS (LHC style BPM)

Major reworking of codebase (broken API – Memory exhaustion)

OP request to be able to switch between SX.ACQW20 and SIX.W20 to trigger acquisitions

Test done by hand in Q3 – OK

Still need to clean API and add ‘proper’ way to switch acquisition trigger

New SEM grid electronics FESA class

Plenty of debugging and ‘discovery’ at first beam

BGI

New PyQt GUIs (serving also as proof as concept for PyQt)

New Temperature Power and Cooling devices delivered to operations

Systems with Notable changes (SPS)

ALPS

FESA class continuously evolved during LS2 - Worked from 1st beam

Modes gradually implemented and evolved (FIFO, 0ms Orbit, First Turn, Snapshot, Capture, etc)

API already defined with OP in 2019 but evolved during LS2

BPMSPS_PHASER

New class to contain the main logic for the phasing and act as data source to the expert applications

Expert Tools agogo (Jordi ++)

Scaling factor management, Calibration control, ...

Interlocks (extraction etc) - Continuous discussion with OP

BFCTSR

Firmware changes for intensity measurement reflected in FESA

Systems with Notable changes (SPS)

dBLM

KT from Lars

Finally merged with LHC class

BESTLD

Class started by Lars but redesigned to better fit use-cases

Added spill read-out

New task for PS EA using extra electronics for pulse output (vs LTIM)

BESCLD (SEM-cloud acq in BA1)

Class made by Lars early LS2 replaced by BESTLD (easier long-term maintenance)

Systems with Notable changes (LEIR AD ELENA)

BPMADLNA

Extensive refactor after G. Baud handover

Moved to MBLT (30% speed increase solving data bottleneck)

New algorithm to approximate peak-to-peak

Class extended to cover ELENA Schottky and AD

BSCRAELE (Scraper for ELENA)

New FESA class from scratch

BPMLI

New FESA class LEIR Injection BPM

New Acquisition and calibration functionality etc.

Systems with Notable changes (LEIR AD ELENA)

BPMLE

Departure of Lars and Lars left a gap in knowledge
FESA class not in a good state – Needed redesign
(Physical!) Hardware problems (SIS3300 VMOD-TTL)
New documentation made to capture knowledge

Systems with Notable changes (LHC)

BOBR

Moved to EDGE2

Careful migration of many instances and config cleanup

BpmFipSgt (BPM/BLM FIP control)

API and functionality extended to include BLM specific reset capabilities

Renovated BLM Fip Reset GUI

Additional logging

Di/Dt

New FESA class for readout etc

BPMLHC

Migration of codebase towards baseline FESA

Ping-pong application resurrected and readied for several RF ball tests

Adapted to the run3 dependencies & used extensively during LS2 with minor issues solved

Systems with Notable changes (LHC)

BLMLHC

BLM Threshold GUIs major re-hash to fit with CO changes

Basic FESA3 version already deployed in 2019 so Greek team could work

Significant redesign of FESA2 class - New MBLT allowed faster readout etc

Consolidation during 2020 of expert tools needed for teams in BL

Sequencer tasks optimized (still to be deployed)

A lot of work from many people

BQLHC

New FESA class implementing LHC specific functionality

Uses new BQBBQ class as core

Now allows any number of devices for various tune-fitting algorithms

BWSLHC

Many issues identified and resolved (during 2021!)

Old FESA code replaced with new code from Injector wire scanners

Systems with Notable changes (LHC)

BCTDCLHC2 (16 Bit BCT)

Code was massively simplified – Some errors during simplification

Missing MTT functionality

Reimplemented calibration routines and sequencer tasks

BST

New tools (GUIs) for experts

DOROS

Issues with mode switching (glitches) solved in collaboration with IQ

Deployment of more collimator devices – New FEC

Assistance given to Collimation team to set up new devices

Still discovering missing KT from Guillaume (Capture etc)

Systems with Notable changes (LHC)

Injection BPM

KT from Lars

Code not changed much, but many problems needed ironing out

Inconsistencies with FECs & CCDE

LUMENS issues

VME windows vs DABInit for firmware loading

dB LM

New system

Saw beam during test

New modes (Capture etc)

To be commissioned correctly during 2022

Systems with Notable changes (AWAKE CLEAR)

COTS BPM system from iTech (CLEAR BPMs)

Served as a (painful) proof-of-concept exercise of tackling a COTS system in the accelerator control system

Took considerably longer than you would expect !

Lib dependency problems etc

Various fixes & adding of new AWAKE 'burst mode'

BPM-AWAKE - Awake p+ BPMs - FESA class + Expert GUI

BPM-AWAKE-LEC - Awake Common Beam Line BPMs- FESA class + Expert GUI

BTV-STREAK - Awake Streak Cameras - FESA class + Expert GUI

BTV-AWAKE-SPEC - Awake Spectrometer - FESA class + 3 Expert GUIs

Systems with Notable changes (ISOLDE)

Scanners

Adapt for new motors/ mechanics

Reworked calculation of beam position

Faraday Cup

Increase of PLC communication from 2Hz to 10Hz

HI ISOLDE Beam Diagnostics Boxes

Reviewed (configuration) + new installations

FESA adapted to new firmware from PM

TOF + Digitizer

New functionality

All events capture with energy histogram

Systems with Notable changes (Experiment Areas)

Over a dozen FESA classes stripped back and redone (almost) from scratch

BXCET, BXDWC, BXBPF, BXCED, BXFISC, BXMWPC, BXSCAL, BXSCINT, etc...

Many classes had 'hidden' technical dept within them

e.g. Instead of porting properly to FESA old (very old) code in standalone binaries, the FESA classes were simply making a system call to a binary!

Complete overhaul of Expert GUIs which were rather poorly supported in the past

Systems with Notable changes (Cross-domain)

BWSLIU

Interface for operation already implemented during 2019

Position and profile simulation data from Matlab files

Real data from SPS and PS to let the operational applications develop

New fit library was introduced

BWSLIUEXP class interfaced with hardware

→ VME → eventually IPBUS

Procedure for the movement came at end of 2020 and rapidly implemented in FESA

BWSACQ class read-out of raw and integrated data + settings

Legacy Wire scanners

FESA class revived

Debugging with new VMA card needed for MEN CPU

Peculiarities coming in Linux vs old LYNXOS

Already deployed in calibration bench during 2020

Allowing calibration of scanners before re-installation

Final debugging with timing before re-installation into machine

Systems with Notable changes (Cross-domain)

BOSTEP

Revived and will be maintained until all MIDI motors go to BstepMotorVME (Enrico Bravin)

17 devices still in BOSTEP

BTVDC

Generic FESA class to cover all new installations of Basler cameras

Expanded to cover all installations from simple cases (AD, etc) to SBDS to AWAKE

AWAKE is a new frontier for our FESA classes – Huge deployment of FESA class (64 core machine, 20 cameras)

SBDS employs extra algorithms for saturation detection

Systems with Notable changes (Cross-domain)

BLMINJ

First deployment was ready for L4 LBE run

Then deployed all way to TT2/TT10

Maintained the ultimate goal of a single FESA class from the LINAC to TT10

With additional BLMSYNC and BTCInterlock classes to cover PS 10Hz and L4 tail-clipper

BQBBQ

BQSB DAB based system rolled out for ELENA

New VFC based BQBBQ class replacing BQSB/BQBBQLHC

Satisfies all need of all accelerators

Flexible enough to provide data in PPM machine + AD, LHC etc

Introduced FESA functions instead of aligned arrays for settings

Performance problems in PSB and LEIR now disappeared

Systems with Notable changes (Cross-domain)

BOPLCIO and or BOSEMCTRL (SEMGrids PSB PS & ISOLDE)

Common classes homogenizing machines and different instrument parts by design

Systems with Notable changes (General)

SIS3302

Needed for several BI systems but didn't work out the box
SBP collaboration with D. Cobas to debug for BI's usage

DABInit

Code finally compiled in L867

Work on deploying new version in coming weeks

Recap on Major New FESA classes

ALPS

BTVDC

BWSLIU

BQBBQ

BFC (+IQ+OP)

DBLMs

SEMs

Systems Requiring Significant (non-coding) Efforts

DOROS

Limited KT (without beam!) after G. Baud departure – Lack of documentation on several aspects

Understanding how hardware behaves a challenge for SW and IQ

BLMLHC

Zoo of applications

Java Fx threshold application dependencies needed to be re-understood

Rediscovering sequencer tasks

BPMLHC

Limited KT (without beam!) after G. Baud departure

Loss of Lars also meant we were truly alone for the run 3 restart

Nevertheless, no delay for October tests

Contingencies - Just in Case...

MOPOS was kept on standby in case of ALPS problems

PPC, LYNXOS, old NFS server, etc

Significant work for CO and BI at start of LS2

Involved creation of new BPMOPOSGW gateway class

Bridging between PPC island and post LS2 control system

BGV was kept on standby in case demonstrator was needed

Significant KT on 'how it works' and how to resurrect post LS2

Thankfully, all no longer required > LS2

GUIs

General

GUI journey has been a bumpy ride

Big Investment during run 2 in JavaFX

Only to be *effectively* abandoned by CO/CSS

Though it will remain supported

By the end of run 2 we had

PyQt (Qt5)

C++ Qt (1 application)

Java Swing / Fx

Maven build and deploy

The beginnings of FESAWeb

Java Swing GUIs

We were in good shape as we generally use common methods and libraries for building GUIs

Common dependencies

Similar code-structure

Nevertheless a lot of work needed to be done on the libraries and per-application code

Adapting to run 3 baselines from CO

RBAC, LSA, JAPC, etc...

Keeping JavaFx Alive

As with Swing, JavaFx also benefited from common libraries

But not as easy to support

Lots of the code was written by departed staff / fellows

Half-hearted support from CO

PyQt GUIs have been made since 2014 in BISW

No common structure

RAD tools rather than long-term GUIs

Investment in PyQt made through trainees and technical students

Proof of concept (Maria+Steen)

Collaboration with CO (Sara+Steen)

Many things added to fulfill BI's needs like RBAC, build and deployment env, etc..

ComRAD came to the party recently

Promises to give us a zero-code solution

Experience is mixed so far

Many issues and bugs found by Sara, Javier, Mathieu, Manuel

Still pursuing and collaborating with CSS to make a usable product for BI

Worry that CSS already seems to be abandoning the efforts

FESAWeb

Zero code, **zero design visualization of any FESA Device/Property/Field**

FESAWeb is not WRAP or Inspector

... zero design remember

Note that FESA Web started the party several years ago

WRAP just arrived !

Born as a response to address many broken Swing GUIs at the end of run 2

Either rewrite again in Swing or ...

Many of these GUIs were recognized as simple fixed-displays

No need for a traditional application

Difficulty in finding new recruits who can make good GUIs in Java and Swing

Bringing GUIs Together at End of LS2

Start to have a zoo of possible GUIs

New Applauncher

Users don't care about how it's implemented but need a one-stop launch tool to launch any GUI

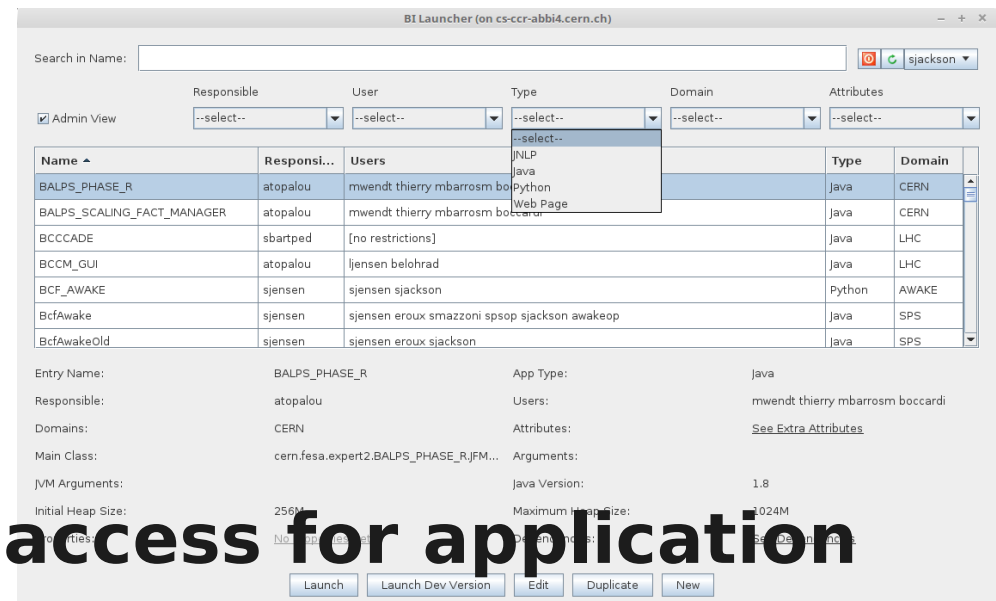
Java

PyQt

ComRAD

FESAWeb

CBNG



Also Incorporates RBAC access for application protection

Common Environment

BI-SW also assist group with general computing environment

Complete shift to VMs from bdidevX and BITS terminal servers during LS2

Still need common environment to start our applications

CCM config for BI-EXPERT

BI-LIDS renovated

Old LIDS was LHC-centric and obsolete

Manual intervention to update most info

Replaced with **BI Portal**

Move from SVN to Gitlab

All BISW GUIs already in Git

Assistance to other users in the group to move to Git

New tools made as basis for Python development

Aimed at GUIs but can be applied to any Python project

IT Support to the Group

BISW also mandated to assist BI users with general IT

Python development

UCAP

Data Analysis (access to NXCALS, notebooks, PM data access etc)

EOS

Etc...

Towards the Future

Software Technical Debt

The section is forever adding new software and updating tools/classes

Systems rarely get removed!

Hopefully during the next run, we can finally address accumulated technical debt

Classes inherited from John (Areas) are relatively well in control

Classes handed over from departed staff are understood but would benefit from revising

It would also be good to have more than 1 person on some classes

Pair programming, peer reviewing, Agile techniques can produce higher quality results

Comes at a cost in resources – Resources we never had in the past

Harmonizing FESA classes

Many approaches to do the same thing in many classes

As a section, we are organizing adhoc sessions to discuss various aspects of SW

- Encore and EDGe access

- Possible CBNG (3) migration from Maven

- Documentation options and recommendations

- Common math and algorithm libraries

- Etc...

Hopefully we can converge on a common recommended way to implement things for new or renovated projects during run 3

Continuing Homogenization

BLMINJ, BWS, BBQ, dBLM, BOPLCIO, BOSEMCTRL, BTV are all good examples of how we successfully homogenized systems

Single FESA class

Hardware and software teams only have to have knowledge of 1 type of system

Single code-base to maintain in the future

Other classes can also benefit from this?

BCT, more dBLM, BPMs, ...?

Continuing Homogenization

Homogeneous classes are not always the answer though

The generic BLMINJ class, took almost as long as writing 5 unique classes!

But BL section now benefits from a single class to use in future

Some generic classes brought lots of complex configuration

Less classes, but much *heavier* classes to make work!

Another approach is to homogenize the building blocks for several independent classes

EA classes all follow same approach after renovation by Stephane/Arak/Joe

Look into extending some ideas into other classes in future

One Last Word....

Inter-section collaboration worked well during LS2 but...

For some systems, specifications came very late

If people in hardware sections have 3 or 4 projects to do in LS, can we at least not start the kickoff for all projects at the start of an LS?

Remember that BISW are not *just a service section*

We are as much part of the project/team as everyone else

We should avoid statements like 'FESA class and API late'

As we know, if this is the case, it is almost always because the project ran late!

Questions