Trip Event Logger - EuXFEL – Maxwell HPC Cluster – AI

Connect a laser to a super computer and have it controlled by artificial intelligence.

EuXFEL
DOOCS
Control System
OPC-AO
EPICS
(TANGO)

Trip Event Logger (TEL)
Online Analysis

ChimeraTK
Application Core
Analysis Module

TRIPEVENT
Reports
States

DOCCS DAQ Snap
Sample reduced
MCS DAQ Data

Not yet automated

Data Sets
For long term use

-Labled data
-Trainings data
-Simulated Data
.raw (MCS DAQ)
.hdf5

Ladybug
Offline Module
Analysis adapter
for **HPC**
command line tool

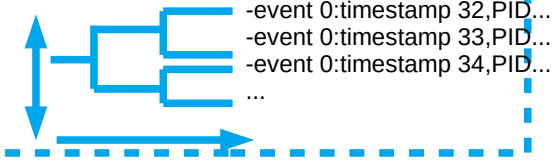
Other scripts
Docker container
File and
data set manger

Analysis Modules

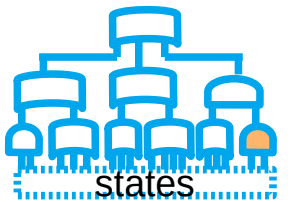
- **C++**
- *MATLAB*
(-PYTHON)
-…

setData(&raw)
setData(&ttf2_daq_getdata)
GetParameter()
Calculate()
WriteResults("results.h5")
Evaluate() → state

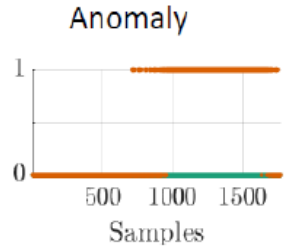
Fault State Collector
-Event tree
(global state of all modules
at the moment of first transition
to a fault)
Not yet implemented



Fault Tree
Event display
Failure probability calculation
Not yet implemented



Detailed Results

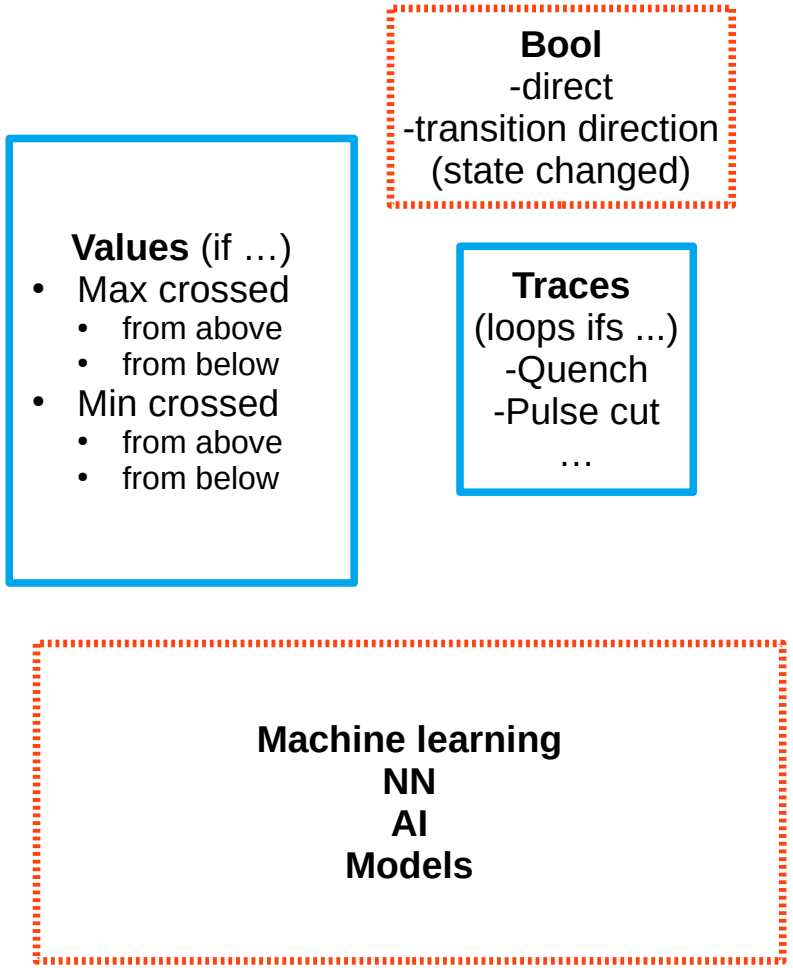


DESY. Contact: Jan Timm - DESY MSK - jan.horst.karl.timm@desy.de

Analysis module

C++ written analysis libraries. Each represents an event / component in the Fault/Event Tree

- Input in our case are the 6 cavity signals for cavities (for other components this could also be traces, scalar or bools).
  - But also meta data and parameter about the operation mode of the machine if necessary.
- The output could also be a traces, scalars etc.
  - e.g. written in hdf5 for further investigation, especially in the case of offline analysis,
  - but you must also evaluate the output for the TEL and break it down to at least 3 **states**: off, normal, (warning1 .. warningN), failure for **Fault/Event Tree Analysis**.
- Each module must know its location, time stamp.
- For offline and online analysis.



Trip Event Logger

It should detect failures and than report them with a failure telegram.

Telegrams are collected, as for fault tree analysis and for collecting data of interest.

- Telegram: failure state, location, time, (link to data, failure probability).
- ChimeraTK Application Core
  - This not only enables **monitoring**, but also allows you to **intervene** in the control system.
  - There are several control system backends. (DOOCS, OPC-AO, EPICS, *(TANGO)*)
  - Thread management is under the hood.
  - Scalable, if enough computing power is available.
- A model based analysis of cavities will be part of this soon (and the first module), developed by Ayla Nawaz.