Monitoring Tools for the Future

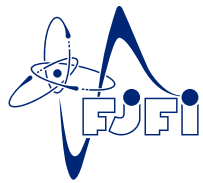
Adjustment of COOOL and MurphyTV to the new FriDAQ protocol

Aleš Suchomel

Czech Technical University in Prague

Faculty of Nuclear Sciences and Physical Engineering

DAQFEET



Introduction

MurphyTV

Presents organized tables of data acquisition error statistics.
Scope: ≈ 5 000 lines
Tools: C++, ROOT (GUI), qmake
Repository: RCCARS

COOOL

Creates and displays detector-specific histograms of physical data.
Scope: ≈ 30 000 lines
Tools: C++, ROOT, bash, make
Repository: CORAL (standalone)

DaqDataDecoding

Converts raw bit streams into maps of objects suitable for subsequent processing. Identifies data errors.
Scope: ≈ 20 000 lines
Tools: C++, Expat, bash, make/cmake
Repository: CORAL

Goals

1. Analyze data quality monitoring tools MurphyTV and COOOL.
2. Design a way to adapt these tools to the new FriDAQ protocol.
3. Improve their original functionality whenever possible.
4. Implement, test and deploy proposed changes.

Feasibility

MurphyTV

Focuses on service data → decoding physical data is unnecessary.
Impacted: ≈ 10 000 lines
Result: Nothing to save, ok scale
⟶ **Reimplement completely**

Needed resources:

- FriDAQ format description
- FriDAQ data samples
- Original MurphyTV
- User consultations

State: In Progress

COOOL

Analyzes physical data → interpretation of all data words is essential.
Impacted: ?? (up to 50 000 lines)
Result: Full rewrite very demanding
⟶ **Reuse whatever possible**

Needed resources:

- Data input definition (post-HLT?)
- Identified detector-specific changes
- Original COOOL
- Detector expert involvement

State: Postponed

FriMurphyTV - Requirements

- **Functional** $\rightarrow$ **Restore**
  - Decode, analyse and visualize error information from FriDAQ headers.
  - Enable asynchronous and continuous data processing.
- **Technological** $\rightarrow$ **Update**
  - Drop unnecessary dependencies (DaqDataDecoding, ROOT).
  - Use uniform technologies with the rest of the RCCARS.
  - Support Linux-based OS exclusively.
- **User Experience** $\rightarrow$ **Preserve**
  - Keep consistent UI whenever possible.
  - Extend focus on a specific detector.
- **Other** $\rightarrow$ **Streamline**
  - Ensure high performance of both data decoding and analysis.

Contact us in case of a request exceeding the original functionality.

FriMurphyTV - Infrastructure

Technologies

Core: C++, Qt Objects
GUI: QML, Qt Widgets
Build: qmake, GCC
Tests: GoogleTest
Version control: Git, GitLab
Platform: CentOS 7

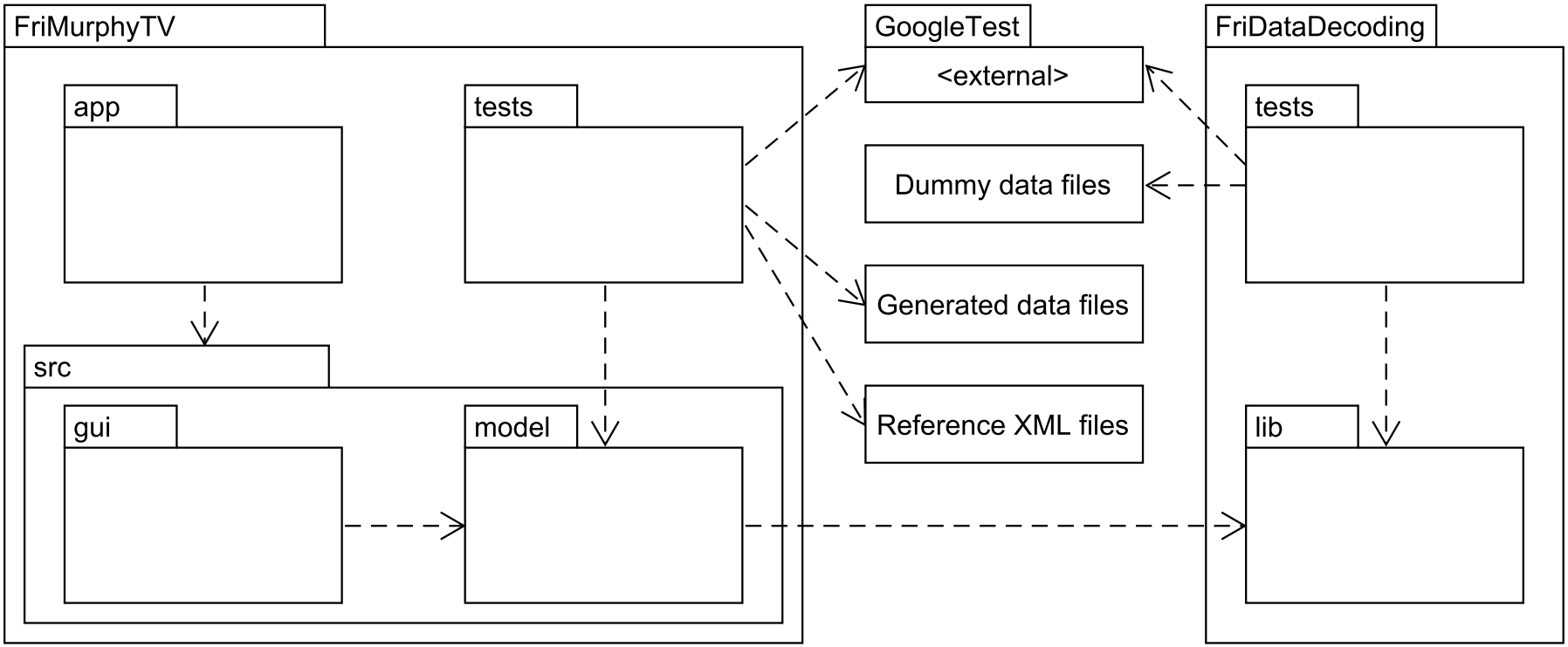
Dependencies

- Lightweight FriDAQ decoder
- FriDAQ data generator
- DIALOG

Other

- FriDAQ protocol description

FriMurphyTV – Architecture Design



Conclusion

- **MurphyTV**
  - Should be rewritten to the new FriDAQ protocol within this project.
- **COOOL**
  - Problematic as it requires decoding and interpretation of all detector-specific data. More clarification is needed.
  - Reimplementation would be very demanding. Transforming new data into DaqDataDecoding API whenever possible seems more feasible.

Thank you for your attention!

Contact: suchoale@fjfi.cvut.cz