DUNE DAQ Project

DUNE Doc.: DUNE-D-XX-NNNN
EDMS Id: 1234567

DEEP UNDERGROUND
NEUTRINO EXPERIMENT

DAQ Project Specification Document

Trigger and Data AcQuisition (TDAQ) System Specifications

Abstract

The TDAQ system is responsible for the acquisition and selection of data produced by the DUNE detectors, as well as for their synchronisation. Different instances of it will be initially used at DUNE test stands such as ProtoDUNE, for detector prototypes characterisation and validation. Two independent, almost identical TDAQ instances will serve the horizontal drift technology (FD1-HD) and the vertical drift technology (FD2-VD) far detector modules. This document describes the specifications for the TDAQ system and its components.

DUNE DAQ Project	
DUNE Doc:	DUNE-D-XX-NNNN
EDMS Id:	1234567
EDMS Url:	`https://edms.cern.ch/document/1234567/1`
Version:	0.2
Created:	August 16, 2021
Last modified:	August 26, 2021

Prepared by:	**Checked by:**	**Approved by:**
G. Lehmann Miotto, A. Thea, ...	FD1-HD technical board, FD2-VD technical board	DAQ Consortium management board

[INTENTIONALLY BLANK PAGE]

Revision History

Revision	Date	Description
0.1	15/08/2021	Initial draft.
0.2	25/08/2021	Added text for data collection.

Table of Contents

Revision History . . . iii

Table of Contents . . . iv

1 Introduction . . . 3

- 1.1 Purpose and Scope of Document . . . 3
- 1.2 Document Overview . . . 3
- 1.3 Conventions and Glossary . . . 3
- 1.4 Related Documents . . . 7

2 DUNE Far Detector TDAQ description . . . 8

- 2.1 Requirements summary . . . 10
- 2.2 Description of the DUNE Far Detector DAQ . . . 10
- 2.3 External DAQ Interfaces . . . 15
- 2.4 DUNE Far Detector DAQ components . . . 15
  - 2.4.1 Timing and Clock Distribution . . . 15
  - 2.4.2 Readout . . . 16
  - 2.4.3 Trigger and Data Filter . . . 17
  - 2.4.4 Data Collection . . . 19
  - 2.4.5 Run Control, Configuration and Monitoring . . . 19
  - 2.4.6 Data Quality Monitoring . . . 20
- 2.5 Physical Description . . . 20

3 Detailed Functional Description and Specification . . . 21

- 3.1 Timing . . . 21
  - 3.1.1 Interfaces . . . 21
  - 3.1.2 Specifications . . . 21
- 3.2 Readout . . . 21
  - 3.2.1 Interfaces . . . 21
    - 3.2.1.1 Facility . . . 21
    - 3.2.1.2 CCM . . . 21
    - 3.2.1.3 Data collection . . . 22
    - 3.2.1.4 Trigger . . . 22
  - 3.2.2 Specifications . . . 23
- 3.3 Trigger . . . 23
  - 3.3.1 Interfaces . . . 23
    - 3.3.1.1 Trigger Inputs . . . 24
      - 3.3.1.1.1 Upstream DAQ . . . 24
      - 3.3.1.1.2 Hardware Signals Interface (HSI) . . . 24

3.3.1.1.3 External Trigger Interface 24
3.3.1.2 Trigger Outputs 24
3.3.1.2.1 Data Flow 24
3.3.1.3 Controls and Monitoring 24
3.3.1.3.1 Configuration 24
3.3.1.3.2 Data Quality Monitoring 24
3.3.2 Specifications 24

3.4 Data collection 24
3.4.1 Interfaces 24
3.4.1.1 Readout 25
3.4.1.2 Trigger 26
3.4.1.3 Data Filter 26
3.4.1.4 CCM 27
3.4.1.5 Offline Computing 27
3.4.2 Specifications 28

3.5 Data filter 28
3.5.1 Interfaces 28
3.5.1.1 Inputs 28
3.5.1.1.1 Data Collection: 28
3.5.1.2 Outputs 29
3.5.1.2.1 Data Collection: 29
3.5.1.2.2 Data Quality Monitoring: 29
3.5.2 Specifications 29

3.6 Dataflow libraries 29
3.6.1 Interfaces 29
3.6.1.1 Application framework 29
3.6.1.2 Inter process messaging 29
3.6.1.3 Data and message formats 29
3.6.2 Specifications 29
3.6.2.1 Application framework 29
3.6.2.2 Inter process messaging 29
3.6.2.3 Data and message formats 29

3.7 CCM 29
3.7.1 Interfaces 29
3.7.1.1 Control 29
3.7.1.1.1 Configuration 30
3.7.1.2 Monitoring 30
3.7.1.2.1 Operational monitoring 30
3.7.1.2.2 Logging 30
3.7.2 Specifications 31
3.7.2.1 Access 31
3.7.2.2 Control 31
3.7.2.3 Configuration 32
3.7.2.4 Monitoring 32
3.7.2.4.1 Operational monitoring 32
3.7.2.4.2 Logging 32

3.8 DQM 33
3.8.1 Interfaces 33
3.8.2 Specifications 33

3.9 Programming Model 33

4 Development model . . . 34
5 Testing, Validation and Commissioning . . . 35
6 Reliability Matters . . . 36
  6.1 Consequence of Failures . . . 36
  6.2 Prior Knowledge of Expected Reliability . . . 36
  6.3 Measures Proposed to Insure Reliability of Component and/or System . . . 36
  6.4 Quality Assurance to Validate Reliability of Design and Construction or Manufacturing Techniques . . . 36
  6.5 Quality Control to Validate Reliability Specifications during Production . . . 36
References . . . 37
References . . . 38
7 Specification Document Appendices . . . 38
References . . . 39
8 Requirements and their classification, Specification rules . . . 39
  8.1 Requirements . . . 39
  8.2 Recommendations . . . 40
  8.3 Guidelines . . . 40
  8.4 Remarks . . . 40
  8.5 Definitions . . . 41
  8.6 Specification Rules . . . 41
References . . . 43
9 Tables. . . . 43

Preface

[REMOVE THIS CHAPTER FROM YOUR DOCUMENT]

What is expected from a Specification Document?

A specification often refers to a set of documented requirements to be satisfied by a material, design, product, or service. A specification is often a type of technical standard.

There are different types of technical or engineering specifications (specs), and different usages of the term in different technical contexts. They often refer to particular documents, and/or particular information within them. The word specification is broadly defined as "to state explicitly or in detail" or "to be specific".

A Requirement Document is a set of documented requirements, to be satisfied by a given material, design, product, service, etc. It is a common early part of engineering design and product development processes. This is what we've documented in the URD. A functional specification is a kind of requirement, and may show functional block diagrams.

A design or product specification describes the features of the solutions for the Requirement Document, referring to either a designed solution or final produced solution. It is often used to guide fabrication/production. Sometimes the term specification is here used in connection with a data sheet (or spec sheet), which may be confusing. A data sheet describes the technical characteristics of an item or product, often published by a manufacturer to help people choose or use the products. A data sheet is not a technical specification in the sense of informing how to produce.

An "in-service" or "maintained as" specification, specifies the conditions of a system or object after years of operation, including the effects of wear and maintenance (configuration changes).

In the Specification Document we shall document and demonstrate how the Requirements have been analysed and how they will be satisfied in the design of a system, sub-system or of a component.

[extracted and modified from `https://en.wikipedia.org/wiki/Specification_(technical_standard)`*]*

What is the difference between requirements and specifications?

A Requirement Document addresses the "what", and Technical or Design Specification Document addresses the "how". In a perfect world, Requirements precede the Design Specification, and each should be written by different people with different knowledge and viewpoints.

Under solicitation, stakeholders communicate the requirements that the product/service/facility shall be designed to meet through the set of requirements in the Requirement Document. This provides guidance to the design team.

The Requirement Document should state what the various stakeholders of the product need in terms of features, functions, performance, constraints, and quality, written in terms of what the product must do or qualities it must have. These stakeholders, including customers, users, maintenance, tech support and others, have the knowledge to define these requirements. Sometimes these stakeholders state their requirements in terms of design, often unintentionally. There are requirement best practices for correcting this problem and for giving the designers the latitude to define the best solution.

The Specification Document reflects a possible implementation design and provides directions to the builders and coders of the product. Through this document, designers communicate the design for the product to which the builders or coders must comply.

The Design Specification shall state how the design will meet the requirements. Design is not a one-to-one response to requirements. Design requires discipline knowledge and integration of disciplines in most cases. Design will be concerned with trade-offs – between different approaches to meet the requirements and concerning issues such as build or buy. The Design Specification will contain information about the product architecture and describe how each component will contribute to meeting the requirements.

[extracted and modified from `https://reqexperts.com/resources/requirements-articles/articles-what-is-the-difference/`]

1 Introduction

This chapter should summarise very briefly in a first section the scope of the document and of the system/sub-system/component being specified, referring at most to the TDR only as a conceptual design, identifying its main functions and elements, its stakeholders, and the associated user requirement document, which should also be listed in Section 1.4.

It should also contain sections with conventions and glossary

1.1 Purpose and Scope of Document

This document describes the specifications for the TDAQ system, deriving from the overall DUNE physics and far detector modules requirements.

1.2 Document Overview

The document is organised into five chapters. In chapter 1, the glossary and relevant external documents are listed. In chapter 2 an overview of the TDAQ system is given, summarising the main requirements and constraints, outlining the external system interfaces and presenting the different TDAQ components. Chapter 3 contains the detailed functional description and specification for each TDAQ component. Chapters 4 and 5 describe the TDAQ development model and the means by which the system is tested and validated.

1.3 Conventions and Glossary

one-pulse-per-second signal (1PPS signal) An electrical signal with a fast rise time and that arrives in real time with a precise period of one second

analog-to-digital converter (ADC) A sampling of a voltage resulting in a discrete integer count corresponding in some way to the input

advanced mezzanine card (AMC) Holds digitizing electronics and lives in Micro Telecommunications Computing Architecture (μTCA) crates

anode plane assembly (APA) A unit of the FD1-HD detector module containing the elements sensitive to ionization in the liquid argon (LAr). It contains two faces each of three planes of wires, and interfaces to the cold electronics and photon detection system

ASIC application-specific integrated circuit

baryon number A quantity expressing the total number of baryons in a system minus the number of antibaryons

baryon-number violating (BNV) Describing an interaction where baryon number is not conserved

DAQ control, configuration and monitoring subsystem (CCM) A system for controlling, configuring and monitoring other systems in particular those that make up the data acquisition (DAQ) where the CCM encompasses RC

cold electronics (CE) Analog and digital readout electronics that operate at cryogenic temperatures

conventional facilities (CF) Pertaining to construction and operation of buildings and conventional infrastructure, and for the LBNF and DUNE enterprise (LBNF/DUNE), CF includes the excavation caverns

cluster on board (COB) An ATCA motherboard housing four RCEs

commercial off-the-shelf (COTS) Items, typically hardware such as computers, that may be purchased whole, without any custom design or fabrication and thus at normal consumer prices and availability

charge-parity symmetry violation (CPV) Lack of symmetry in a system before and after charge and parity transformations are applied. For CP symmetry to hold, a particle turns into its corresponding antiparticle under a charge transformation, and a parity transformation inverts its space coordinates, i.e. produces the mirror image

charge readout (CRO) The system for detecting ionization charge distributions in a detector module

charge-readout plane (CRP) An anode technology using perforated PCB-based charge readout (CRO) with projecting electrodes immersed in LAr

data acquisition (DAQ) The data acquisition system accepts data from the detector front-end (FE) electronics, buffers the data, performs a trigger decision, builds events from the selected data and delivers the result to the offline secondary DAQ buffer

DAQ front-end computer (DAQ FEC) The portion of one DAQ partition that hosts the DAQ data receiver (DDR), DAQ primary buffer and data selector. It hosts the DAQ front-end readout (FER) and corresponding portion of the DAQ primary buffer

DAQ front-end fragment The portion of one DAQ partition relating to a single DAQ front-end computer (DAQ FEC) and corresponding to an integral number of detector units. See also data fragment

DAQ partition A cohesive and coherent collection of DAQ hardware and software working together to trigger and read out some portion of one detector module; it consists of an integral number of DAQ front-end fragments. Multiple DAQ partitions may operate simultaneously, but each instance operates independently

DAQ readout subsystem (DAQ RO) The subsystem of the DAQ for accepting and buffering data input from detector electronics

DAQ trigger subsystem (DAQ TS) The subsystem of the DAQ responsible for forming a trigger decision

data fragment A block of data read out from a single DAQ front-end fragment that span a contiguous period of time as requested by a trigger command

data selection The process of forming a trigger decision for selecting a subset of detector data for output by the DAQ from the content of the detector data itself. Not to be confused with data selector

data selector The portion of the DAQ front-end fragment that accepts trigger commands and returns the corresponding data fragment. Not to be confused with data selection

detector unit A portion of a far detector module may be further partitioned into a number of similar parts. For example, the FD1-HD time projection chamber (TPC) is made up of anode plane assembly (APA) units (and other elements)

DP module dual-phase DUNE far detector (FD) module

data quality monitoring (DQM) Analysis of the raw data to monitor the integrity of the data and the performance of the detectors and their electronics. This type of monitoring may be performed in real time, within the DAQ system, or in later stages of processing, using disk files as input

Deep Underground Neutrino Experiment (DUNE) A leading-edge, international experiment for neutrino science and proton decay studies; refers to the entire international experiment and collaboration

DUNE timing and synchronization subsystem (DUNE DTS) The portion of the DAQ that provides for timing and synchronization to various detector systems

external trigger interface (ETI) Interface between module trigger logics (MTLs) and external source and sinks of relevant trigger information

external trigger logic (ETL) Trigger processing that consumes far detector module level trigger notification information and other global sources of trigger input and emits trigger command information back to the MTLs

external trigger candidate Information provided to the MTL about events external to a far detector module so that it may be considered in forming trigger commands

far detector module The entire DUNE far detector is segmented into four modules, each with a nominal 10 kt fiducial mass

far detector (FD) The 70 kt total (40 kt fiducial) mass liquid argon time-projection chamber (LArTPC) DUNE detector, composed of four 17.5 kt total (10 kt fiducial) mass modules, to be installed at the far site at Sanford Underground Research Facility (SURF) in Lead, SD, USA

horizontal drift technology (FD1-HD) LArTPC design in which electrons drift horizontally to wire plane anodes (anode plane assemblies) that along with the front-end electronics are immersed in LAr

far detector module 2 (FD2) The second far detector module to be built at SURF

vertical drift technology (FD2-VD) LArTPC design in which electrons drift vertically to PCB-based anodes at the top and bottom of the LAr volume, with a cathode in the middle

front-end (FE) The front-end refers to a point that is “upstream” of the data flow for a particular subsystem. For example the FD1-HD front-end electronics is where the cold electronics meet the sense wires of the TPC and the front-end DAQ is where the DAQ meets the output of the electronics

front-end mother board (FEMB) Refers a unit of the single-phase (SP) cold electronics (CE) that contains the FE amplifier and analog-to-digital converter (ADC) ASICs covering 128 channels

Fermi National Accelerator Laboratory (Fermilab) U.S. national laboratory in Batavia, IL. It is the laboratory that hosts Deep Underground Neutrino Experiment (DUNE) and serves as its near site

field programmable gate array (FPGA) An integrated circuit technology that allows the hardware to be reconfigured to execute different algorithms after its manufacture and deployment

FS (1) The far site, SURF, where the DUNE far detector is located; (2) “Full stream” relates to a data stream that has not undergone selection, compression or other form of reduction

Far Site Conventional Facilities (FSCF) The conventional facilities (CF) at the DUNE far detector site, SURF, including all detector caverns and support infrastructure

Global Positioning System (GPS) A satellite-based system that provides a highly accurate one-pulse-per-second signal (1PPS signal) that may be used to synchronize clocks and determine location

liquid argon (LAr) Argon in its liquid phase; it is a cryogenic liquid with a boiling point of 87 K and density of 1.4 g/ml

liquid argon time-projection chamber (LArTPC) A TPC filled with liquid argon; the basis for the DUNE FD modules

Long-Baseline Neutrino Facility (LBNF) Long-Baseline Neutrino Facility; refers to the facilities that support the experiment including in-kind contributions under the line-item project. LBNF is responsible for developing the neutrino beam, the far site cryostats and far and near site cryogenics systems, and the conventional facilities for DUNE

LBNF and DUNE enterprise (LBNF/DUNE)	Long-Baseline Neutrino Facility/Deep Underground Neutrino Experiment; refers to the enterprise or program including LBNF/DUNE-US and DUNE
main communications room (MCR)	Space at the FD site for cyber infrastructure
module trigger logic (MTL)	Trigger processing that consumes detector unit level trigger command information and emits trigger commands. It provides the external trigger logic (ETL) with trigger notifications and receives back any external trigger candidates
PCB	printed circuit board
photon detector (PD)	The detector elements involved in measurement of the number and arrival times of optical photons produced in a detector module
photon detection system (PDS)	The detector subsystem sensitive to light produced in the LAr
RC	Depending on context, one of (1) resistive-capacitive (circuit), (2) run control, the system for configuring, starting and terminating the DAQ, or (3) resource coordinator, a member of the DUNE management team responsible for coordinating the financial resources of the project
reconfigurable computing element (RCE)	Data processor located outside of the cryostat on a cluster on board (COB) that contains field programmable gate array (FPGA), RAM and solid-state disk (SSD) resources, responsible for buffering data, producing trigger primitives, responding to triggered requests for data and synching supernova neutrino burst (SNB) dumps
readout window	A fixed, atomic and continuous period of time over which data from a far detector module, in whole or in part, is recorded. This period may differ based on the trigger that initiated the readout
signal-to-noise (S/N)	signal-to-noise ratio
secondary DAQ buffer	A secondary DAQ buffer holds a small subset of the full rate as selected by a trigger command. This buffer also marks the interface with the DUNE Offline
supernova neutrino burst (SNB)	A prompt increase in the flux of low-energy neutrinos emitted in the first few seconds of a core-collapse supernova. It can also refer to a trigger command type that may be due to this phenomenon, or detector conditions that mimic its interaction signature
SuperNova Early Warning System (SNEWS)	A global supernova neutrino burst trigger formed by a coincidence of SNB triggers collected from participating experiments
single-phase (SP)	Distinguishes a LArTPC technology by the fact that it operates using argon in its liquid phase only
solid-state disk (SSD)	Any storage device that may provide sufficient write throughput to receive, both collectively and distributed, the sustained full rate of data from a far detector module for many seconds
Sanford Underground Research Facility (SURF)	The laboratory in South Dakota where the DUNE FD will be installed and operated; also where the Long-Baseline Neutrino Facility (LBNF) Far Site Conventional Facilities (FSCF) and the FS cryostat and cryogenic systems will be constructed
TAI	International Atomic Time
Micro Telecommunications Computing Architecture (μTCA)	The computer architecture specification followed by the crates that house charge and light readout electronics in the DP module
TDAQ	trigger and DAQ system
TDR	Depending on context, either “technical design report,” a formal project document that describes the experiment at a technical level, or “technical design review,” a formal review of the technical design of the experiment or of a component

time projection chamber (TPC) A type of particle detector that uses an E field together with a sensitive volume of gas or liquid, e.g., LAr, to perform a 3D reconstruction of a particle trajectory or interaction. The activity is recorded by digitizing the waveforms of current induced on the anode as the distribution of ionization charge passes by or is collected on the electrode (TPC is also used for "total project cost")

trigger candidate Summary information derived from the full data stream and representing a contribution toward forming a trigger decision

trigger command Information derived from one or more trigger candidates that directs elements of the far detector module to read out a portion of the data stream

trigger decision The process by which trigger candidates are converted into trigger commands

trigger notification Information provided by MTL to ETL about trigger decision processing

trigger primitive Information derived by the DAQ FE hardware that describes a region of space (e.g., one or several neighboring channels) and time (e.g., a contiguous set of ADC sample ticks) associated with some activity

warm interface board (WIB) Digital electronics situated just outside a FD cryostat that receives digital data from the front-end mother boards (FEMBs) (part of CE) over cold copper connections and sends it to the reconfigurable computing element (RCE) FE readout hardware

White Rabbit (WR) A component of the timing system that forwards clock signal and time-of-day reference data to the master timing unit

1.4 Related Documents

In this section all interface documents with external systems are listed, for reference.

JT COMP and JT DAQ/SC Consortium Interface Document. URL: `https://edms.cern.ch/document/2145151/3`.

JT DAQ/SC and JT CAL/CI Consortia Interface Document. URL: `https://edms.cern.ch/document/2088741/3`.

SP PDS and JT DAQ/SC Consortia Interface Documents. URL: `https://edms.cern.ch/document/2088726/3`.

SP TPC and JT DAQ/SC Consortia Interface Document. URL: `https://edms.cern.ch/document/2088713/4`.

2 DUNE Far Detector TDAQ description

The trigger and DAQ (TDAQ) system is responsible for receiving, processing, and recording data from the DUNE experiment. This system:

- provides timing and synchronization to the detector electronics and calibration devices;
- receives and buffers data streaming from the TPC and the photon detection system (PDS);
- extracts information from the data at a local level to subsequently form trigger decisions;
- builds trigger records, defined as a collection from selected detector space-time volumes corresponding to a trigger decision;
- carries out additional data reduction and compression as needed; and
- relays trigger records to permanent storage.

The main challenge for the DUNE TDAQ lies in the development of effective, resilient software and firmware that optimize the performance of the underlying hardware. The design is driven not only by data rate and throughput considerations, but also – and predominantly – by the stringent uptime requirements of the experiment.

The TDAQ has been subdivided into a set of subsystems. Figure 2.1 shows the different subsystems and their relationships. All subsystems rely on the functionality provided by the *CCM* and *DQM*, that is the glue of the overall TDAQ, transforming the set of components into a coherent system. The *DAQ trigger subsystem (DAQ TS)* and *Data filter* are in charge of the selection and compression of data. The *Dataflow* subsystem provides the communication layer to exchange data (i.e., it is used by the other subsystems intersecting it in the diagram). In addition it implements the data collection functionality, i.e., the logic for building trigger records as well as the organization of data into files. The *DAQ readout subsystem (DAQ RO)* receives the data streams from the TPC and PDS, processes them to extract information for the trigger and buffers data while the trigger is forming a decision. The *DUNE timing and synchronization subsystem (DUNE DTS)* is in charge of distributing the clock, synchronizing the far detector module, as well as timestamping hardware signals that may be used for triggering, such as calibration pulses. The *Detector electronics software* is not part of the TDAQ responsibility: it is shown in the diagram to illustrate the fact that detector experts will develop the software to configure, control, and monitor the electronics using the tools provided by the *CCM* subsystem.

The TDAQ is mainly composed of commercial off-the-shelf (COTS) components, as shown in Figure 2.2. A high performance Ethernet network interconnects all the elements and allows them to operate as a single, distributed system. At the output of the TDAQ the high-bandwidth Wide Area Network (WAN) allows the transfer of data from the SURF to Fermi National Accelerator Laboratory (Fermilab).

The TDAQ for the whole of DUNE FD is designed and developed coherently by a joint team. The TDAQ systems for the different FD modules differ only in minor details so as to support the electronics and the data selection criteria for each.

In particular, the TDAQ systems for the FD1-HD and FD2-VD detector modules will be very similar, with customization for the anode planes and the photon detector (PD) electronics readout, as well as for the data selection algorithms, which will be tuned to the modules' geometry. The main TDAQ features for the FD are described in the DUNE FD TDR [1, 2].

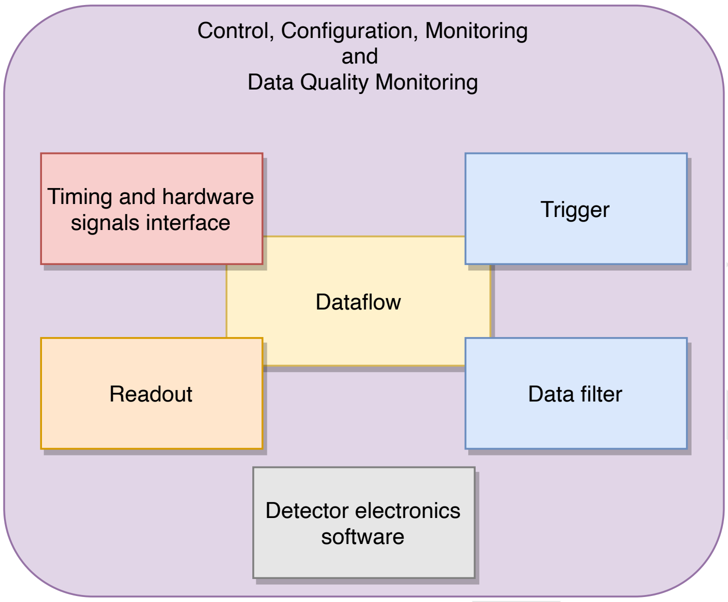

Figure 2.1: Diagram showing the relationships between TDAQ subsystems. It also shows, in grey, the detector electronics software, which is not part of TDAQ but will be developed fully embedded into the CCM environment.

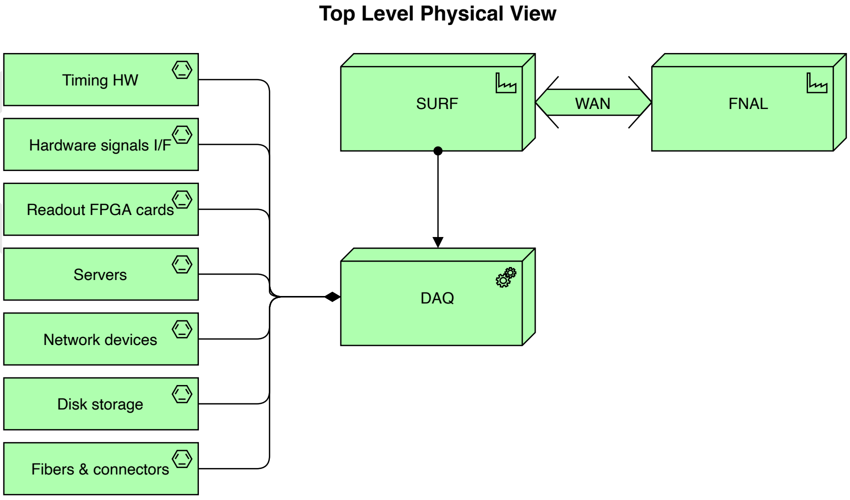

Figure 2.2: Physical view diagram of the TDAQ system. At SURF, the hardware components are distributed across the detector caverns and the surface DAQ room. The TDAQ data are transferred to Fermilab over a Wide Area Network (WAN) connection.

2.1 Requirements summary

The specifications for the TDAQ deriving from higher-level requirements [4] are listed in Table 2.1. These specifications can be viewed as falling into four categories:

- Synchronization:
  - The timing system shall provide a common timestamp to all FD detector systems.
  - It shall be able to align the timestamp across FD systems, the TPC and PDS, within one FD to better than 10 ns at all times.
  - It shall be able to distribute synchronization, calibration, and control commands to FD systems.
- Data selection:
  - The FD shall be > 90% efficient for any interaction that leaves > 100 MeV of visible ionization energy inside the fiducial volume.
  - The DAQ TS shall be capable of recognizing a SNB from a nearby supernova based on a threshold of more than 60 interactions within 10 seconds, with neutrino energy deposition above 10 MeV each, with an efficiency >95%.
  - The FD shall have high efficiency for any interaction leaving < 100 MeV of visible ionization energy inside the fiducial volume, and for single interactions with visible energy deposit > 10 MeV. The lowest-energy of these may be pre-scaled.
- Data throughput:
  - The TDAQ shall be able to receive digitized data from the detector electronics and buffer them for up to 10 seconds awaiting a trigger decision.
  - It shall be able to store data for up to one week without interrupting the DAQ, in case of delays in the data transfer from SURF to Fermilab.
  - It shall provide enough disk I/O capacity for driving the WAN link at 100 Gb/s.
  - It should be able to transfer SNB trigger records (140 TB (FD1-HD) and 180 TB (FD2-VD)) from the detector caverns to the DAQ components on the surface within one hour (≈700 Gb/s).
- Uptime: Each FD module shall have an uptime of at least 95%, and the FD as a whole shall have an uptime of at least 98%, during which at least one module is operational. Since a few days per year of downtime for infrastructure maintenance cannot be avoided, and individual hardware failures (electronics, servers, disks, ...) may occur, this translates into a very strong requirement for the TDAQ. The TDAQ shall operate continuously, dynamically adjust to changing conditions, tolerate faults, and recover from errors autonomously.

The electronics parameters essential for dimensioning the TDAQ system are shown in Tables 2.2 and 2.3. The indicative number of TDAQ components for the each far detector module are shown in Tables 2.2 and 2.3. The individual TDAQ component counts for the FD1-HD and FD2-VD are shown in Tables 2.4 and 2.5.

2.2 Description of the DUNE Far Detector DAQ

The FD TDAQ system is physically located at the FD site, SURF. It uses space and power in each underground detector cavern (on top of the cryogenics mezzanine), and above-ground, in the main communications room (MCR) within the Ross Dry building. The upstream part of the system, responsible for raw detector data reception, buffering, pre-processing, and triggering, resides underground. The back-end, which is responsible for trigger-records-building, data storage, and data filtering, resides on the surface. Some elements of the TDAQ span both locations, i.e., the timing, control, configuration, and monitoring systems (DUNE DTS and CCM). The inter-connectivity between all TDAQ elements is provided through a distributed and redundant

Table 2.1: TDAQ specifications.

Label	Description	Specification (Goal)	Rationale	Validation
SP-FD-22	Data rate to tape	$<$ 30 PB/year	Cost. Bandwidth.	ProtoDUNE
SP-FD-23	Supernova trigger	Efficiency for a SNB producing at least 60 interactions with a ν energy >10 MeV in 12 kt of active detector mass during the first 10 s of the burst.	$>$95% efficiency for SNB within 20 kpc	Simulation and bench tests
SP-DAQ-1	DAQ readout throughput: The DAQ shall be able to accept the continuous data stream from the TPC and Photon detectors.	1.4 (FD1-HD), 1.8 TB/s (FD2-VD) per single phase detector module	Specification from TPC and PDS electronics	Modular test on ProtoDUNE; overall throughput scales linearly with number of APAs
SP-DAQ-2	DAQ storage throughput: The DAQ shall be able to store selected data at an average throughput of 10 Gb/s, with temporary peak throughput of 100 Gb/s.	10 Gb/s average storage throughput; 100 Gb/s peak temporary storage throughput aggregated over all detector modules	Average throughput estimated from physics and calibration requirements; peak throughput allowing for fast storage of SNB data ($\sim 10^4$ seconds).	ProtoDUNE demonstrated steady storage at $\sim$ 40 Gb/s for a storage volume of 700 TB. Laboratory tests will allow to demonstrate the performance reach.
SP-DAQ-3	DAQ readout window: The DAQ shall support storing triggered data with a variable size readout window, from few μs (calibration) to 100 s (SNB), with a typical readout window for triggered interactions of 2.6 (FD1-HD) and 4.25 ms (FD2-VD) respectively.	10 μs < readout window < 100 s	Storage of the complete dataset for up to 100 s is required by the SNB physics studies; the typical readout window is defined by the drift time in the detector; calibration triggers can be configured to readout data much shorter time intervals.	Implementation techniques to be validated on the ProtoDUNE setup and in test labs.
SP-DAQ-4	Calibration trigger: The DAQ shall provide the means to distribute time-synchronous commands to the calibration systems, in order to fire them, at a configurable rate and sequence and		Calibration is essential to attain required detector performance comprehension.	Techniques for doing this have been run successfully in MicroBooNE and ProtoDUNE.

Table 2.2: Summary of FD1-HD detector parameters driving the DAQ design.

Parameter	Value
TPC channels	384000
TPC channel count per APA	2560
TPC electronics 10 G links	1500
TPC ADC sampling rate	1.953125 MHz
TPC ADC dynamic range	14 bits
PDS channels	??
PDS electronics 4.8 G links	150
Max localized event record window (1 drift length)	2.6 ms
Extended event record window	100 s
Representative size of TPC localized event record (uncompressed, full detector for 1 drift length)	2.5 GB
Full size of TPC extended event record (uncompressed)	140 TB

Table 2.3: Summary of FD2-VD detector parameters driving the DAQ design.

Parameter	Value
TPC channels	491520
TPC channel count per CRP	3072
TPC top electronics 40 G links	320
TPC bottom electronics 10 G links	960
TPC ADC sampling rate	1.953125 MHz
TPC ADC dynamic range	14 bits
PDS channels	1280
Max localized event record window (1 drift length)	4.25 ms
Extended event record window	100 s
Representative size of TPC localized event record (uncompressed, full detector for 1 drift length)	8 GB
Full size of TPC extended event record (uncompressed)	180 TB

Table 2.4: TDAQ component counts for FD1-HD. All servers are interconnected via 10/100 G Ethernet network.

Parameter	Count
Clock speed	$62.5\,MHz$
Timing endpoints	≈ 300
TPC readout cards	150
PDs readout cards	15
Readout servers	85
Trigger servers	20
Data collection, storage servers	15
Data filter servers	20
TDAQ control and monitoring servers	25

Table 2.5: TDAQ component counts for FD2-VD. All servers are interconnected via 10/100 G Ethernet network.

Parameter	Count
Clock speed	$62.5\,MHz$
Timing endpoints	??
TPC readout cards	80+80?
PDs readout cards	40?
Readout servers	120
Trigger servers	20
Data collection, storage servers	15
Data filter servers	20
TDAQ control and monitoring servers	25

10/100 G Ethernet network. Data flow through the DAQ from upstream to the back-end and then offline. Most raw data are processed and buffered underground, thus controlling consumption of available data bandwidth to the surface.

All the detector modules and their subsystems are synchronized and timed against a common global clock, provided by the DUNE DTS. Cross-module communication and communication to the outside world for data selection (trigger) purposes is facilitated through an external trigger interface (ETI).

The TDAQ system must acquire data from the both the TPC and the PDS in order to satisfy the requirements placed on it that derive from the experiment's three main physics objectives: (1) measuring neutrino charge-parity symmetry violation (CPV) and related long-baseline oscillation parameters using the high-intensity neutrino beam from Fermilab; (2) measuring atmospheric neutrinos and searches for rare processes such as baryon-number violating (BNV) decays; and (3) detecting neutrinos from a nearby SNB. Ionization charge measurement by the TPC for any given activity in the FD requires a nominal recording of data over a time window determined by the drift speed of the ionization electrons in LAr and the detector dimension along the drift direction (6.5 m). Given a target drift E field of 450 V/cm, the time window is set to 4.25 ms. The activity associated with beam, cosmic rays, and atmospheric neutrinos is localized in space and particularly in time; SNB neutrinos are associated with activity that extends over the entirety of the detector and lasts between 10 and 100 s.

The detector data flow from the electronics output links, on top of the cryostat, through the TDAQ system to permanent storage at Fermilab. Two stages of data selection allow reduction of the overall data volume from ≈4.7 PB/year produced by the FD2-VD module to ≈30 PB/year for all FD modules: triggering the collection of data only for interesting detector regions and time windows, and applying data compression algorithms. Data selection strategies are defined in close collaboration with the DUNE physics groups, the offline processing team, and the detector subsystem and calibration experts.

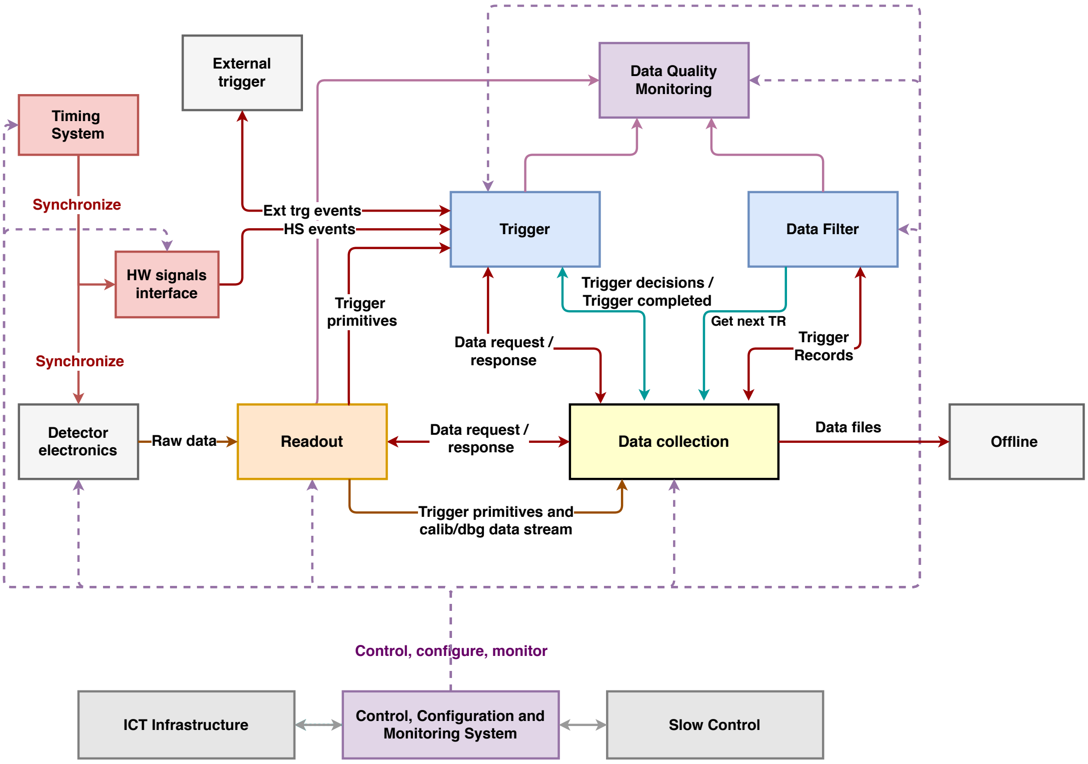

Figure 2.3: Conceptual overview of TDAQ system functionality for a single FD module. External systems are depicted in grey while the TDAQ subsystems are represented using the same color scheme as in Figure 2.1. The external interfaces are described in Section 2.3 while the flow of data and messages is described in Section 2.2. Acronyms used in the figure: HW=hardware, HS=hardware signal, TR=trigger record, ICT=information and communication technologies

Figure 2.3 depicts the flow of data within the TDAQ system, left to right. Raw data produced by the detector electronics is streamed into the readout (DAQ RO). Here data are processed to produce trigger primitive information, and stored for up to 10 s. The trigger subsystem (DAQ TS) receives trigger primitives, timestamped hardware signal events, and external trigger events, and combines this information to form a trigger decision. Trigger decision messages containing a list of time windows and detector locations are forwarded to the data collection component; data collection is in charge of requesting the relevant data from the DAQ RO and the DAQ TS, forming trigger records, and storing them to disk. Trigger records are dispatched by data collection to the data filter for further data reduction. The data filter returns the modified trigger records to the data collection component for both storage and transfer to the offline computing system at Fermilab. The whole DAQ process is orchestrated by the CCM; the quality of the raw data is continuously monitored by the DQM (Section 2.4.6).

In the following sections the different TDAQ components shown in Figure 2.3 and the external TDAQ interfaces are described in more detail.

2.3 External DAQ Interfaces

The overview diagram in Figure 2.3 shows the main external interfaces of the TDAQ, with the external systems depicted in grey.

The DUNE DTS (Section 2.4.1) distributes the clock and unique timestamp to the detector electronics, including any calibration devices.

The hardware signals interface injects timestamped hardware signals, e.g., from calibration systems, into the TDAQ data selection chain.

The DAQ RO (Section 2.4.2) receives raw data from the various detector electronics devices.

The DAQ TS (Section 2.4.3) receives external timestamped trigger messages (e.g., from other FD modules or the SuperNova Early Warning System (SNEWS)) to initiate data collection irrespective of any activity inside the module.

The data collection (Section 2.4.4) interfaces to the offline computing, for the purpose of transferring raw data and metadata for permanent storage.

The CCM (Section 2.4.5) interfaces with detectors and calibration devices, providing the software suite that allows these components to be controlled, configured and monitored by the TDAQ. The CCM also exchanges data with the slow control system and the Information and Communication Technologies (ICT) infrastructure monitoring system.

2.4 DUNE Far Detector DAQ components

2.4.1 Timing and Clock Distribution

The DUNE timing and synchronization subsystem (DUNE DTS) provides services to allow the accurate synchronization of sampling and data processing for all elements of the FD. The primary function is the distribution of electrical and optical signals to FE electronics, such that every data sample may be tagged with a 64-bit timestamp. These DUNE DTS timestamps, which are unique across the lifetime of the experiment, are used for data processing and event building within the TDAQ, and are included within the recorded data sample to support event reconstruction within and across detector modules, and the correlation of DUNE data with external events (e.g., beam spills or astronomical observations).

All DUNE electronics is synchronized to a common clock, derived from a single Global Positioning System (GPS)-disciplined source. Individual detector channels are synchronized to within a fraction of the 8 ns timestamp granularity. Due to the extended physical size of the DUNE detectors, consistency of timing alignment is ensured by both hardware- and software-based feedback loops, calibrating propagation delays on the timing links. The requirement on the absolute accuracy of timestamps with respect to TAI is one microsecond, though in practice is expected to be much better.

Table 2.6: TDAQ components external interfaces.

TDAQ Component	Interface description
Timing	put short interface descr.
Readout - TPC cold electronics	9.6 G point-to-point links using FELIX FULLMODE 8/10 bit encoded protocol
Readout - FD2 top TPC electronics	100G switched ethernet using UDP protocol
Readout - PDS	4.8 G point-to-point links using FELIX FULLMODE 8/10 bit encoded protocol
Trigger - external trigger sources	Ethernet input of timestamped signals from calibration devices, other FD modules or SNEWS
Data collection - offline computing	WAN ethernet connection for transfer of HDF5 raw data files.
CCM - offline computing	databases with the archive of run configurations and conditions
CCM - slow control	Ethernet based inter process communication to exchange status information
CCM - detectors software	Software libraries and tools for the implementation of the control, configuration and monitoring of the detector electronics

The top drift FE digital electronics has an embedded timing distribution system, based on the White Rabbit (WR) standard, which, starting from common GPS-derived timing signals, distributes the synchronization signals and the DUNE DTS timestamps to the digitization units in the μTCA crates (Section **??**). This system also performs an automatic compensation of the propagation delays so that all the end-node digitization units are aligned to the 1PPS signal coming from the GPS system with sub-ns accuracy.

The bottom drift electronics, and the PDS, use a similar timing distribution system to the FD1-HD module. This relies upon a passive optical fan-out/fan-in of timing signals to several hundred instances of FE electronics. This system also implements an automatic compensation of propagation delays, based on measurement of round-trip time.

The consistent timing synchronization of all detector channels is continuously monitored and checked at multiple levels: within the timing distribution systems themselves; within the DAQ system against calibration signals; and offline against physics signals. As critical infrastructure for data-taking, the timing distribution systems are designed for robustness and fault-tolerance. All electronic components of the system for which failure would affect a large detector volume have redundant components contained in separate crates with redundant power and network links. Two independent GPS systems are used, with antennae at the Ross and Yates shaft entrances respectively, and are able to operate in a hot-spare configuration.

2.4.2 Readout

The DAQ readout subsystem (DAQ RO) is responsible for receiving data from the electronics. It interfaces with the top and bottom drift TPC electronics and the PDS electronics via optical fibers that connect the detector electronics on the cryostat roof to the DAQ FE readout units on the cryogenics mezzanine, inside the DAQ barrack.

Each top-drift electronics CRO μTCA crate aggregates raw data from 10 advanced mezzanine cards

(AMCs) into a 40 Gbit/s optical link to the DAQ, for a total of 640 charge-readout plane (CRP) channels per crate, and of 400 links to the DAQ FE readout. The typical effective throughput of the data link is expected to be around 17 Gbit/s.

The interface between the DAQ and the bottom drift volume electronics is formed by the optical links from the warm interface boards (WIBs), which transfer the raw TPC data to the DAQ FE readout. A total of 1040 10 Gbit/s links is expected; the typical effective throughput of a data link is around 7 Gbit/s.

The DAQ FE readout units store the raw data stream from detectors in a circular memory buffer for at least 10 s, and perform a first stage of preprocessing. Readout units respond to data requests from the event builder by providing the data fragments corresponding to the selected readout window from the temporary memory buffers. In the event an SNB trigger firing, the readout units will stream the content of the temporary memory buffer to local storage for a total duration of 100 second. Afterwards, each readout unit makes the stored SNB record available to the data collection system for transfer to the surface storage.

Readout units perform the first stage of trigger processing on incoming data to generate local elementary trigger information (see trigger primitives in 2.4.3), to minimize the raw data transfer over network. The trigger primitives are streamed to both the DAQ TS for further processing, and to data collection for semi-permanent storage.

In addition to handling the main data path, DAQ FE readout provides extra functions to support debugging, calibration and quality assessment, specifically it:

- supports the bottom-drift CE calibration with a dedicated raw data stream during calibration cycles;
- generates a reduced raw data stream for a configurable subset of channels, on demand, for debugging purposes; and
- samples and extracts basic DQM information.

2.4.3 Trigger and Data Filter

This section briefly summarizes the strategies that have been devised by the TDAQ team to maximize the retention of interesting data while respecting the data volume that can be permanently stored long-term for the DUNE FD (~30 PB/y). The goal is to be as inclusive as possible; at high energies (> 100 MeV) it is important to accept all possible events with as high an efficiency and as wide a region-of-interest in channel and time space as possible, regardless of event type. As energies approach 10 MeV, when radiologicals (including neutron captures, ^{42}Ar, and ^{39}Ar pileup) become dominant, the system should be semi-inclusive, leveraging the topological capabilities of the TPC data to provide some discrimination of low-energy physics signals.

Above 100 MeV of visible energy deposit, the trigger rates are entirely dominated by cosmic rays, which at DUNE's depth will be roughly 4000/day/10 ktonne. The trigger rate at this threshold from neutrino beam events, or atmospherics, is roughly 1000 *per year/10 ktonne*, and thus is a negligible contribution to the overall data volume. At low energies, around 10 MeV of visible energy, backgrounds from radiologicals will begin to dominate. While at the scale of DUNE it is not entirely known precisely what all the low-energy background sources are and what their rates will be, neutron captures on argon, which lead to a 6.2 MeV γ cascade, from neutrons created via (α,n) processes, are expected to be in the range of 1-10 Hz, and will likely dominate until one goes down to 3.5 MeV and hits the 1 kBq expected for ^{42}Ar and its daugther, ^{42}K.

As a reminder, all digitized charge collection data are sent to the readout subsystem and processed. For each strip on a CRP layer, a hit-finding algorithm allows identification of activity above the electronic noise. (With the expected noise of the TPC electronics, the hit-finding threshold is set such that hits will be generated for a large fraction of the ^{39}Ar decays; the spectrum endpoint is at 0.5 MeV.) Every hit generates a so-called trigger primitive.

Similarly, PDS electronics boards send waveform data for any channel that passed an internal threshold; trigger primitives are also formed from these data.

The trigger primitives serve two purposes:

- They are the basic elements used to form a trigger decision in the TDAQ system.
- They are stored as unbiased (at the "event" level) summary information that can be used for trigger, calibration, low-energy physics studies, etc.

To provide good sensitivity to different track toplogies, each trigger primitive contains information such as the time-over-threshold of the waveform, its peak, and its total charge, as well as the timestamp of the start of the waveform. To date, our TPC trigger studies have been done using collection view only; exploitation of induction views is under development but is not seen as needed to satisfy any of DUNE's triggering requirements.

Trigger primitives generated from charge collection channels and the PDS will be stored on disk by the TDAQ system. This data set is very important for carrying out trigger studies but can also be used for calibration purposes, as well as fast data analysis. After compression and minimal clean-up, it is estimated that a few PB/year will be sufficient to store them. It is thus an option for DUNE to not only store the trigger primitives temporarily (a few months) for specific studies, but to make them part of the data that will be stored permanently. This data stream is particularly interesting in view of its role in potentially extending the low-energy physics reach of DUNE beyond its core program; it will also contain summary information for individual interactions with very low visible deposited energy. Depending on the achieved signal-to-noise (S/N) on the collection strips, the trigger primitive threshold is expected to be around 250 keV.

For triggering purposes, the trigger primitives are the basic elements used by the TDAQ to form a trigger record and initiate the collection and storage of raw waveform data. The data selection system takes trigger primitives generated locally and looks for clusters in time and space. These clusters represent what is called "trigger activity." Clusters of trigger activity are then passed to algorithms downstream which determine whether any particular set of trigger activity clusters should be promoted to a trigger candidate. Trigger candidates then are sent to trigger decision logic, which apply criteria that include both configuration parameters (e.g., which triggers are accepted in this data run) as well as dynamic decisions (e.g., does a TPC trigger candidate come after an existing PDS trigger candidate?). The data selection work flow is shown in Figure 2.4.

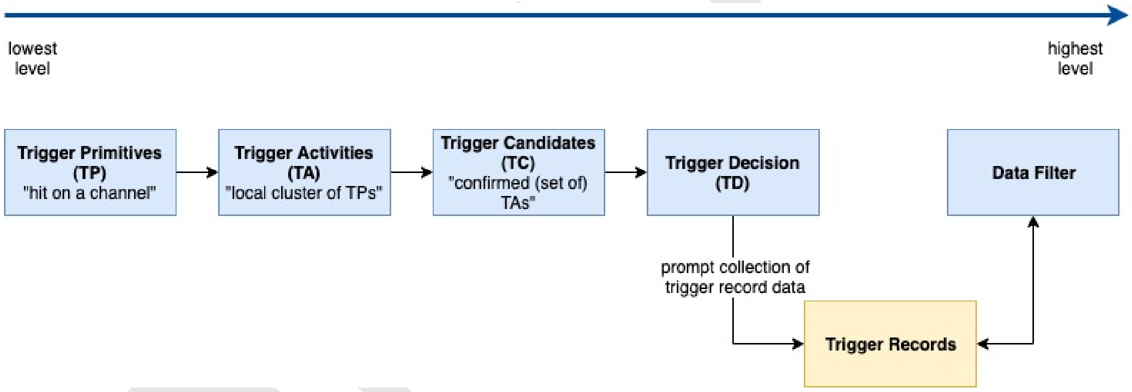

Figure 2.4: The data selection work flow.

There are two different raw data collection modes foreseen for the DUNE FD:

- A trigger decision based on trigger activity consistent with a single interaction or internal decay that includes a list of CRP and/or PD channels to be collected and their associated time window(s). The TDAQ uses this information to collect the relevant raw data from its temporary buffers, form trigger records, and store them persistently. The data files may be further trimmed or compressed through a data filter stage before being transmitted to Fermilab for permanent storage.
- When the trigger identifies several trigger activity clusters within a few seconds that are inconsistent with the expected fluctuations from background in rate and energy, it fires a special trigger decision indicating a SNB candidate. For the FD2-VD the total collected raw data from a SNB will be about ~180 TB of raw data. Thus, while the effective burst threshold must be set low enough to satisfy DUNE's requirements on SNB detection efficiency, it is important to not fire too frequently on background fluctuations. The present assumption is that the trigger conditions will be adjusted such that statistical fluctuations cause on the order of one SNB candidate trigger per month. It will take about one hour to transfer the data from this trigger event from the detector caverns to the storage on surface, and several additional hours to transfer those data to Fermilab. Upon inspection, fake SNB trigger records will be discarded.

Each trigger prompts the collection of data from the DAQ RO to form a trigger record. In the extreme case of an SNB trigger, data from the whole module is collected over a time window of 100 s. In most other cases, data from only a few CRPs and PDs will be collected over much shorter times ($\ll$10 ms).

The data filter acts on already-stored trigger records. It processes them with the aim of further reducing the data volume to be transferred to Fermilab. The techniques considered so far for the data filter include further rejection of trigger records, trimming of the interesting detector regions and time windows within a trigger record, as well as lossless data compression.

2.4.4 Data Collection

The role of the data collection system is to handle the flow of data coherently from their sources to local storage, to provide inputs to data filtering, to collect the filtering result, and to interface with computing for data transfers to permanent storage. It is the only interface to the local DAQ storage at SURF, and is responsible for ensuring consistency, traceability and reliable handling of the data at all times until they are moved to offline computing at Fermilab.

The data collection strategy varies based on the specific readout mode: interaction triggers, SNB triggers, or streaming.

Once an interaction trigger decision is formed the data collection system gathers data fragments from the readout units into coherent trigger records. Intermediate trigger objects that contributed to the trigger decision are added to complement the information. Trigger records are stored on disks prior to being served to the data filter system for further data reduction. The data filter forms modified trigger records that the storage system saves on disk, awaiting their successful transfer to Fermilab.

The SNB records require a different approach due to their size: the data collection system instructs readout units to initiate the raw streaming to local storage on the readout unit. Afterwards, it transfers SNB record fragments to the surface storage servers with high priority, compatibly with the available bandwidth and without interfering with the collection of interaction triggers, which continues in parallel. From the surface storage, the SNB record is transmitted to Fermilab computing over the 100 Gbps WAN link. To minimize the transfer time, which is expected to be a few hours, the data collection system will start transfers as soon as SNB raw data reaches the surface buffer.

The trigger primitives are continuously streamed during data taking. The data collection system is responsible for keeping a copy of the stream in the surface temporary storage for a time period of at least few months. The trigger primitive stream records will be made available on demand.

2.4.5 Run Control, Configuration and Monitoring

The CCM subsystem is in charge of controlling, configuring, and monitoring the TDAQ system, as well as the detector components participating in data taking. It provides a central access point for the highly distributed TDAQ components, allowing them to be treated and managed as a single coherent system though their corresponding subsystem interfaces. It is responsible for error handling and recovery, which is achieved through a robust and autonomous fault-tolerant control system. The main goal is to maximize system uptime, data-taking efficiency and data quality, taking into account that the system will encounter changes in data-taking conditions, both programmatic (e.g., calibrations) and unplanned (e.g., hardware failures or software faults).

The CCM provides an access point that delegates the user's actions, defined as any kind of human interaction, to internal function calls and procedures. It protects the direct access to detector and infrastructural resources. It also controls authentication and authorization, which limits different functionalities to certain groups and subsystems. As an example, only individuals authorized as detector experts can modify the FE configuration through the configuration interfaces, or exclude a CRP from the readout.

The control component validates, distributes and executes commands on the TDAQ, and is in charge of keeping the system in a coherent state. It consists of several components, such as access manager, process manager, resource manager and run control to carry out its tasks and also implements the intelligence required to automatically maintain the system in a properly functioning state or to alert operators if any parts malfunction and cannot be automatically recovered.

The configuration component provides several key elements for the configuration management of the TDAQ components and detector FE electronics. It provides descriptions of system configurations, the ability to define and modify configurations, and graphical user interfaces for the human user to access the data. Data access libraries will hide the technology used for the database's implementation. The configuration component is also responsible for the archiving and bookkeeping of any used configurations.

Highly scalable and efficient operational monitoring is essential during data-taking periods. Any malfunctioning component of the experiment must be identified and reported as soon as possible. The monitoring component is intended to probe the TDAQ components, services, and resources, collect and archive the obtained status information, and provide aggregation and visualization tools.

The types of monitoring information vary greatly, ranging from log/error messages to metrics of different types. The monitoring infrastructure must therefore be flexible enough to seamlessly accommodate additions and modifications, and provide an aggregated view of the system behavior. The monitoring subsystem is a data source for the control subsystem that makes use of the information to automatically optimize the data taking conditions or recover from errors.

2.4.6 Data Quality Monitoring

The DQM subsystem is complementary to the monitoring component of the CCM. Instead of collecting counters, rates and logs, the DQM analyses samples of raw data and compares the results with the expectations, thus assessing the quality of the data themselves, rather than the quality of the DAQ process. The DQM will provide the tools to visualize information and will feed data quality assessment results into the control subsystem, that will alert operators and experts and/or execute automated recovery actions.

The DQM system consists of an infrastructure part that will be designed according to distributed data mining and analysis system techniques. The algorithmic part of the DQM, on the other hand, will continuously evolve as understanding of the detector behavior broadens during commissioning and early operation, and as detector conditions evolve and stabilize.

2.5 Physical Description

The TDAQ system is made of rack mounted COTs components: the timing boards are housed in uTCA crates, while all other I/O devices (readout cards and storage devices) are hosted inside servers. the different elements of the TDAQ are interconnected via 10/100 G ethernet.

For each FD far detector module the timing, readout and trigger hardware are organised within 16 racks in a dedicated, air-cooled barrack on top of the cryogenic mezzanine inside the detector cavern. 100 G up-links connect this part of the TDAQ to a surface counting room, in which 8 further racks are dedicated to data storage and filtering TDAQ equipment, jointly for all far detector modules.

The TDAQ hardware is powered through a UPS system, allowing to avoid short-term power glitches to affect the system.

Each TDAQ barrack underground provides a cooling power of 125 kW, while the available power on surface, dedicated to TDAQ equipment, is limited to 50 kW.

3 Detailed Functional Description and Specification

Divide this chapter into sections depending upon the specifics of the component. Describe the functions and specify pertinent performance specifications. There should be sufficient detail to allow verification of the design prior to fabrication and testing of the fabricated component to verify its proper functionality. If this specification is for a component that integrates other DUNE components or complex commercial components, specific functions that are solely provided by an individual component being integrated can be referred to the underlying component's specification. However, this should not make this document difficult to read. In that case and certainly for all functions shared by more than one component, clear descriptions must be provided in this document even if it means copying descriptions from the underlying specifications into this one. For example, if the integrated component requires programming, it should not be necessary to flip between several underlying specifications in order to program this component.

3.1 Timing

3.1.1 Interfaces

3.1.2 Specifications

3.2 Readout

3.2.1 Interfaces

As shown in Fig. 3.1, the "'readout" subsystem interfaces with several other DAQ subsystems. These are discussed in this subsection.

3.2.1.1 Facility

The readout subsystem is responsible to assist with the readout unit (RU) installation and configuration. The locations of the interfaces are:

- Readout unit (RU) server specification and hardware requirements.
- RU server configuration (BIOS settings, kernel parameters).
- RU I/O card and local storage installation instructions.
- RU I/O card optical fiber routing, topology and mapping of patch panels.

3.2.1.2 CCM

The readout subsystem will use the tools and implement interfaces provided by CCM to configure, monitor, and control the readout I/O cards, readout software modules, and different performance oriented settings of the hosting server. The locations of the interfaces are:

- The control, configuration, and monitoring of I/O cards via software modules.
- Control, configuration, and monitoring of performance related parameters of software components (e.g.: hardware locality).

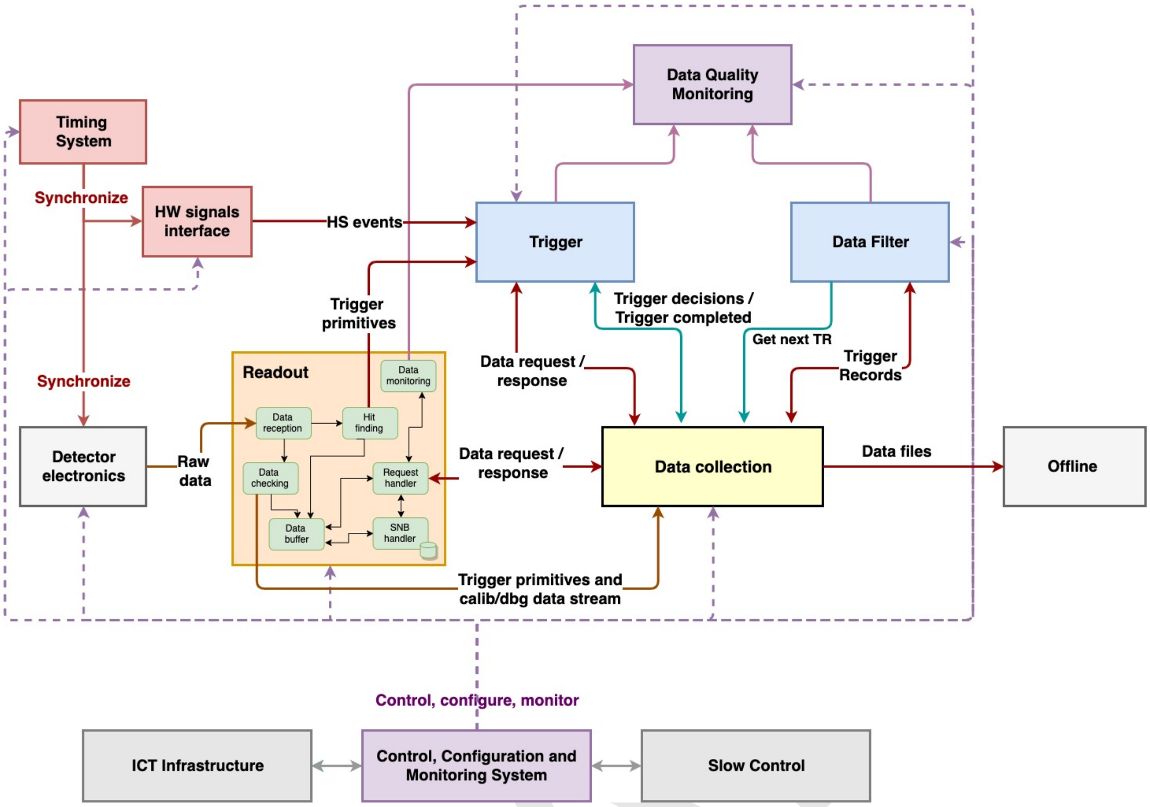

Figure 3.1: Interfaces of Readout components with other DAQ sub-systems.

- Provided parameter list for operational monitoring, and registered variables via operational monitoring interface.
- The control, dynamic configuration, and monitoring of firmware internals (e.g.: link configuration, alignment, adjusting of trigger primitive generation thresholds).
- Possibility for interrupting and resuming the flow of data from the FE to the readout via software.
- Possibility for data integrity inspection on the readout side via software tools.
- Failover, recovery scenarios and instructions of the readout for the control system.

3.2.1.3 Data collection

A detailed description of the readout interface with Data collection is given in the UD/DF interface document. The locations of the interfaces are:

- Requests for triggered data sent by DF to the readout applications owned by Upstream DAQ.
- Requested data sent in a reply by the readout system.
- Data collection requests the readout to begin a supernova burst (SNB) recording.
- Readout notifies Data collection that the SNB data recording is complete.
- Readout handing over ownership of SNB data to Data collection.
- Streaming of trigger primitives into other subsystems or components (Trigger, storage, etc.).
- Streaming of calibration or debug data into other components from the FE.

3.2.1.4 Trigger

A detailed description of the readout interface with Data Selection is given in the UD/DS interface document's "data path" section. The locations of the interfaces are:

- The Trigger specifies maximum trigger latency, maximum readout window and window offset, so the readout system can dimension the hardware parameters and software configurations (latency buffer and request handling) accordingly.
- The type of algorithms to identify relevant data (hits) in TPC and PDS raw data streams are defined by the Trigger system.
- The content and format of Trigger Primitive (TP) data are defined by the Trigger, the implementation of the algorithms within the resources of the readout is the responsibility of the readout system.
- Streaming of trigger primitives into the Trigger modules/applications, using the implementation from the Data collection interfaces.

3.2.2 Specifications

As shown in Fig. 3.2, the readout system consists several components and functional elements.

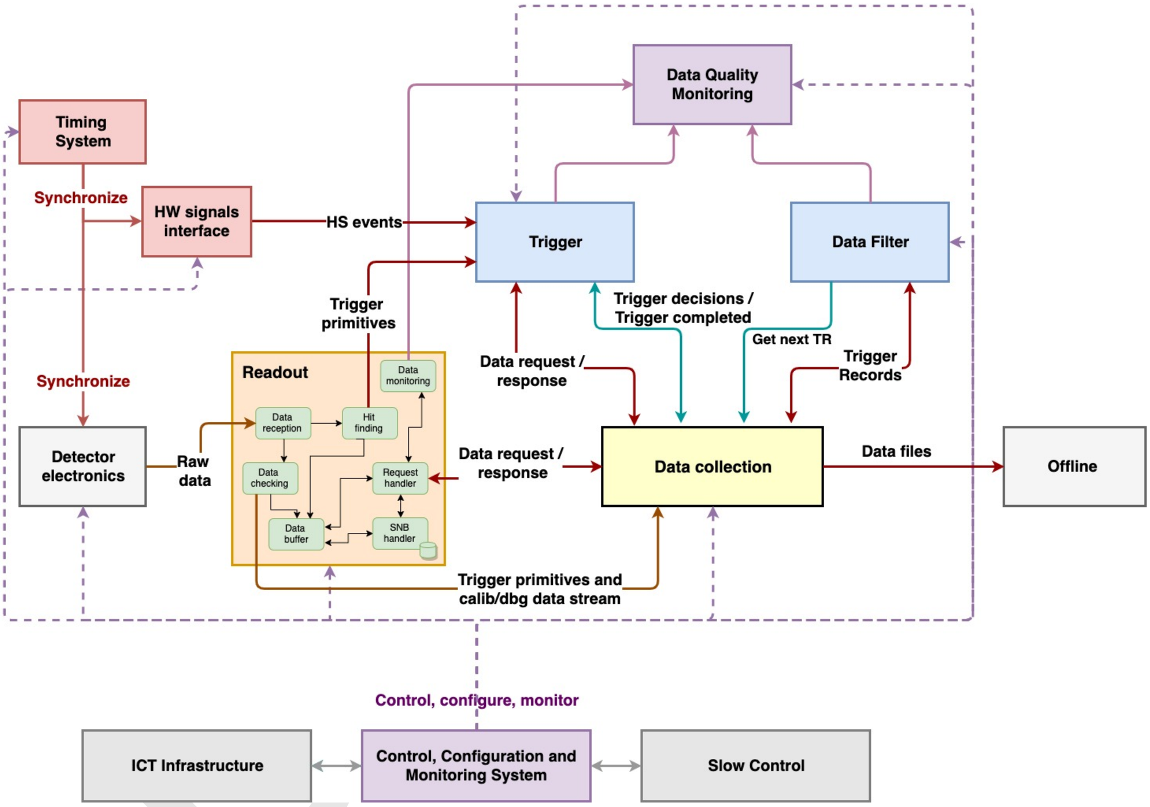

Figure 3.2: Interfaces of Readout components with other DAQ sub-systems.

3.3 Trigger

3.3.1 Interfaces

As shown in Fig. 2.3, the “Trigger” component of the Data Selection system has interfaces to several subsystems, including some that have no other role but to participate in a trigger decision.

3.3.1.1 Trigger Inputs

The trigger takes input from the following subsystems:

3.3.1.1.1 Upstream DAQ The interface with Upstream DAQ provides the primary inputs to the trigger in the form of Trigger Primitives, generated from waveforms from TPC and PDS data.

3.3.1.1.2 Hardware Signals Interface (HSI) The hardware signals interface provides the trigger with Timestamped Data from hardware signals such as those from the calibration system. The Timestamped Data looks to the trigger like Trigger Activity, and gets passed along through the algorithms from there to a Trigger Decision.

3.3.1.1.3 External Trigger Interface : The External Trigger acts to bring in trigger signals from sources outside a single module, such as other DUNE modules, or even other experiments.

3.3.1.2 Trigger Outputs

3.3.1.2.1 Data Flow Once a Trigger Decision (TD) has been made by the Module Level Trigger (MLT), a data request is sent to the Data Flow Orchestrator (DFO) asking that a particular snapshot in time or space, corresponding to the TD, will be saved and sent to disk. The DFO sends acknowledgements of requests back to the MLT, and can also provide back pressure to hold off the MLT from sending new requests. Lastly, DFO requests data from the MLT corresponding to the list of TPs, TAs, and TCs that went into any Trigger Decision.

3.3.1.3 Controls and Monitoring

3.3.1.3.1 Configuration Like most subsystems, the Trigger will be configured by the DAQ CCM. That configuration will include any channels or other front-end components to be masked out from decisions, configurations for calibrations, and trigger logic associated with the current DUNE Run Plan. Operational monitoring of the trigger performance will also be done through CCM.

3.3.1.3.2 Data Quality Monitoring The Trigger provides high-level information to DQM, including the current deadtime accounting (if any), trigger rates broken down by individual trigger types, and other useful diagnostics.

3.3.2 Specifications

3.4 Data collection

3.4.1 Interfaces

The Dataflow system interacts with the following systems:

- The Readout system is the main provider of input to Dataflow. Upon request, Readout sends data associated with triggers in the form of fragments, and it pushes streams of data when configured to do so. For SNB triggers, there are additional messages exchanged between the two systems to start the high-speed storage of data within the Readout system and then later transfer the data to the DAQ storage system.
- The Trigger system sends trigger decision messages to Dataflow that specify the time windows and detector components to be read out for each trigger. Dataflow tells the Trigger when each trigger record has been successfully stored with the DAQ storage system. Upon request, the Trigger sends data associated with triggers in the form of fragments.

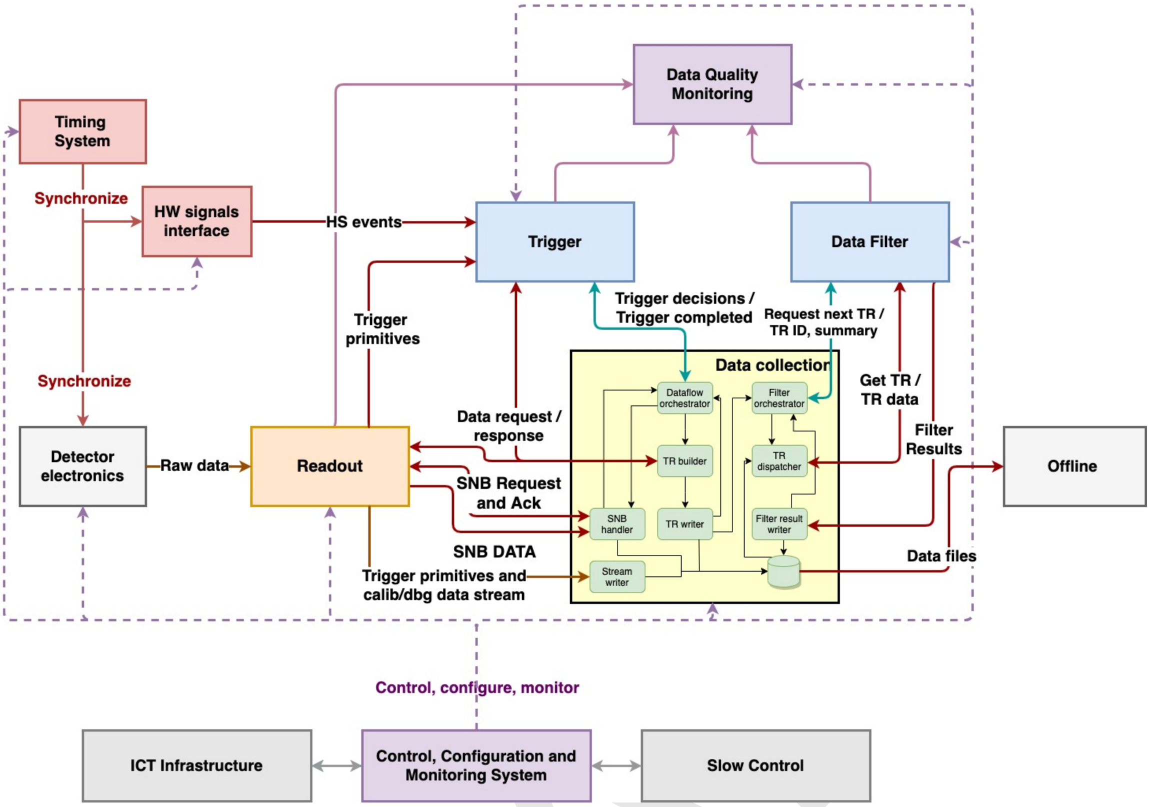

Figure 3.3: Interfaces of Data Collection components with other TDAQ sub-systems.

- The Data Filter system receives trigger records associated with *local* triggers, analyzes them, and returns the results to Dataflow.
- CCM provides services such as configuration, operational monitoring, and message logging.
- Offline Computing is the final destination for the data that is stored.

Questions: 1) "system" or "subsystem"? 2) "Dataflow" or "Data Collection"?
The system interface documents are listed in Table 3.1

Table 3.1: Dataflow interfaces to other systems

System	Specification Documents
Readout	N/A
Trigger	N/A
Data Filter	N/A
CCM	N/A
Offline Computing	N/A

3.4.1.1 Readout

As part of the interaction between the Data Collection and Readout subsystems, the Trigger Record Builder sends data requests for each trigger to the Readout subsystem and receives the requested data fragments in return. Individual data requests are sent to the readout units that are specified in the trigger decision message, and these messages contain the time window of interest for each readout unit. These messages also contain an identifier for the TR Builder that initiated the request so that the fragment replies can be routed to the appropriate TR Builder instance. The fragments that are sent from the Readout subsystem contain the unmodified data that was provided by the detector electronics plus metadata that identifies the source of the fragment, summarizes the contents of the electronics data, and aids in the matching of individual fragments into complete trigger records.

[next, describe the SNB and streaming interfaces]

Remark: useful reference text...

The interface between the Dataflow and readout systems includes the readout of triggered data, the transmission of the continuous stream of trigger primitive data, the notification of a supernova burst trigger and the subsequent readout of that data, and the occasional transmission of continuous streams of limited bandwidth data.

3.4.1.2 Trigger

As part of the interaction between the Data Collection and Trigger subsystems, the Trigger sends trigger decisions to the Dataflow Orchestrator (DFO), and the DFO sends trigger complete messages to the Trigger. It is the responsibility of the Trigger to determine which Readout and Trigger components that should be included in each trigger record, and it conveys that information in the trigger decision message. The Data Collection subsystem collects the data fragments from all of the requested readout units and Trigger processes and writes the assembled fragments as trigger records to the storage system. The DFO informs the Trigger of the disposition of each trigger record (e.g. full or partial success) via the trigger complete message. The trigger complete message is useful to the Trigger system in a number of ways, and those are discussed in the Trigger section of this document.

For the gathering of Trigger data to be stored in trigger records, the Trigger Record Builder sends data requests for each trigger to the Trigger subsystem and receives the requested data fragments in return. Individual data requests are sent to the Trigger process that were specified in the trigger decision message, and these messages contain the time window of interest for each trigger process. The fragments that are sent from the Trigger subsystem contain the information provided by the software processes, plus metadata that identifies the source of the fragment, summarizes the contents of the payload, and aids in the matching of individual fragments into complete trigger records.

3.4.1.3 Data Filter

The Data Filter subsystem initiates the interaction between itself and the Data Collection subsystem by requesting information about the next available trigger record. That message is sent to the Filter Orchestrator. The Filter Orchestrator either returns identifying information for a trigger record or a message that indicates that no unprocessed trigger records are currently available. The Data Filter then uses the identifying information to request the full set of data for each trigger record from the TR Dispatcher, and it then processes the data appropriately (e.g. it runs one or more analysis algorithms on the data, possibly creates data products that should be added to the trigger record, and possibly performs one or more actions on the original data in the trigger record). After completing those actions, the Data Filter sends a message to the Filter Result Writer to indicate how the trigger record should be subsequently handled. The options include discarding the trigger record in its entirety, replacing its original contents with modified contents, and adding information. The baseline plan is for the Data Filter to return the complete set of data to be written to the output storage, but there may be optimizations in which only modified or additional data is sent. The Filter Result Writer writes the appropriate data for each trigger record to the output storage in an area that indicates that it is ready to be copied to offline.

This baseline model for the interaction between Data Collection components and the Data Filter is influenced by several factors, including the following ones:

- Every trigger record from all *local* triggers will be processed by the Data Filter (whereas none of the SNB or streaming data will).
- It is expected that the Data Filter will be interested in all of the data in each of the locally-triggered trigger records, so there isn't a need to provide access to fractional parts of trigger records, at least at the level

of the Data Collection system.

- Local triggers will be a combination of ones that produce lower-rate, larger-size and higher-rate, smaller-size trigger records, so the choice of centralizing the access to the unprocessed trigger records within the Data Collection subsystem allows for optimizations that may be needed in order to handle different trigger record rates and sizes.
- The separation of the data transfer into two steps (fetching information about the trigger records and then requesting the trigger record themselves) will provide the opportunity for the Data Filter to process the trigger records in a slightly different order than what the Filter Orchestrator provides. This may be useful for processing trigger records in a certain priority order within the Data Filter.
- The Data Filter will likely be designed to operate in both online and offline environments, and it is expected that its input and output components will be modularized such that different functionality can be used online and offline while keeping the analysis parts the same.

3.4.1.4 CCM

Messages and information are passed between the Data Collection nd CCM subsystems as shown in Figure 3.4. These interactions include the following:

- CCM sends commands to Data Collection; Data Collection acts on those commands and reports the results of those actions back to CCM.
- CCM sends configuration information to Data Collection at appropriate times. In addition, DC can query CCM for subsets of configuration information and have those returned.
- DC reports the state of its modules and processes to CCM.
- DC sends message strings to CCM for archival, display, and analysis.
- DC sends operational monitoring metrics to CCM for archival, display, and analysis.

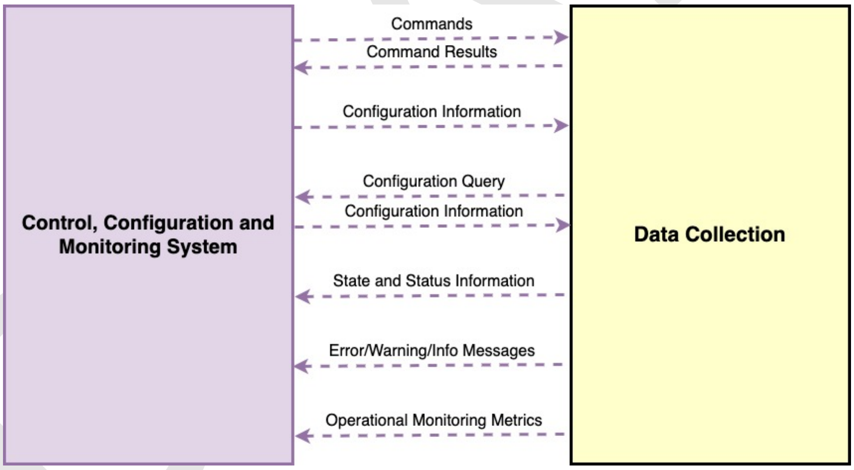

Figure 3.4: Interactions of Data Collection with the CCM sub-system.

In addition to the messages and information that is passed between the systems at run time, there are connections between Data Collection and CCM during development. These take the form of CCM-provided libraries that Data Collection makes us of when communicating with CCM and agreed-upon conventions that developers for both subsystems follow. An example of such a conventions is the use of well-defined exception classes to communicate exceptional conditions between different code units. This allows consistent reporting and reactions to exceptional conditions across the system.

3.4.1.5 Offline Computing

Dataflow prepares data for transfer to offline storage. It makes use of the data-transfer tools that are provided by Offline Computing to ensure that data is transferred reliably.

3.4.2 Specifications

This document provides the specification of the Dataflow system that is being developed for use in the DAQ systems of the Near and Far Detectors. The Dataflow system reads out detector data, packages it appropriately, and stores it in the DAQ storage system. Triggered data, as well as streaming data of contrained bandwidth, are handled by the Dataflow system. Within the DAQ, the Dataflow system delivers data to filter and monitoring systems, and it is responsible for preparing the data for transfer to offline storage.

The Dataflow (DF) system is responsible for moving physics data between components of the DAQ system. In particular, the DF system is responsible for collecting data from readout and trigger applications in response to decisions generated by the trigger, placing that data into files, serving data to the data filter and data monitoring systems, and preparing the overall results for transfer to offline storage.

Figure 3.3 shows the relationship of the Dataflow system with other systems in the DAQ, and it shows the functional areas within the Dataflow system. These functional areas are the following:

- The Dataflow Orchestrator interfaces with the Trigger system and manages which Dataflow processes will handle te readout and storage of each trigger record.
- The TR Builder and TR Writer components do the work of reading out trigger records and storing them. These components may spread their work across multiple processes.
- The SNB Handler interfaces with the Readout system to communicate the occurrence of an SNB trigger and manage the read out of the data.
- The Stream Writer functional area accepts *streaming* data from the Readout system and stores it. Examples of streaming data include the continuous flow of Trigger Primitives or occasional streams of raw wave forms for debugging.

Notes:

- talk about the size of the different parts of the storage system?
- talk about high-performance storage?
- do we want to keep open the option to use alternate technologies like an ObjectStore for the first stage of the storage system?
- describe what sorts of configuration inforamtion will be needed by DC?

3.5 Data filter

The Data Filter processes built Trigger Records to filter events and data to reduce the overall data volume to disk, while retaining the required very high efficiencies for high- and low-energy events. It does not operate on either streamed data, or the full SNB data. The Data Filter differs from the “trigger” part of the overall Data Selection system in two important ways. First, the rate of Trigger Decisions is much lower than the rate of Trigger Candidates. Thus, the Data Filter has time to perform relatively sophisticated processing on Trigger Records. Second, because the Data Filter operates on built Trigger Records, it has access to more information than the more local Trigger Candidates used by the trigger. For example, the Data Filter can use PDS and TPC information together to fiducialize away low-energy background events.

3.5.1 Interfaces

3.5.1.1 Inputs

3.5.1.1.1 Data Collection: As described above in Section **??**, the Data Filter takes as input built Trigger Records provided by Data Collection. These Trigger Records can arrive via a dispatcher, or, if the rate is low enough, from disk storage. The latter option is particularly useful as it will allow Data Filter algorithms to work much like an offline analysis module.

3.5.1.2 Outputs

3.5.1.2.1 Data Collection: As described above in Section 3.4.1.3, the Data Filter passes its filtered data stream back to Data Flow, for final archiving.

3.5.1.2.2 Data Quality Monitoring: The Data Filter provides high-level metrics, derived from its own processing, to Data Flow Monitoring.

3.5.2 Specifications

3.6 Dataflow libraries

3.6.1 Interfaces

3.6.1.1 Application framework

3.6.1.2 Inter process messaging

3.6.1.3 Data and message formats

3.6.2 Specifications

3.6.2.1 Application framework

3.6.2.2 Inter process messaging

3.6.2.3 Data and message formats

3.7 CCM

The CCM provides a set of functionality to control, configure and monitor the DAQ processes that will be used by the subsystems, to control hardware, transport and analyze data.

3.7.1 Interfaces

3.7.1.1 Control

The external CCM interfaces to the control system are defined as follows:

- Interface to users via command line and GUI, providing equivalent access to the functionality list below, and including authentication and authorization.
- Interface to Offline to deliver relevant records of control actions associated with every run
- An interface shall be provided to allow the sending of commands, with configuration data to DAQ processes, and receiving acknowledgement.
- Error handling/monitoring interface to slow control and event data quality monitoring for (automatic) handling of alerts, notifications and scheduled events from these systems.

3.7.1.1.1 Configuration

The external CCM interfaces to the configuration system are defined as follows:

- Interface to users via command line and GUI, providing equivalent access to the functionality list below, and including authentication and authorization.
- Interface between process configuration data schema and configuration editor to provide creation, storage, versioning and audit of configurations for subsystems.
- Interface to offline computing to provide a select, relevant subset of configuration information associated with each run.

3.7.1.2 Monitoring

3.7.1.2.1 Operational monitoring

The CCM is responsible for providing a complete solution for capturing, distributing and archiving operational monitoring information. Operational monitoring information in this context is defined as information related to the operation of the components of the DAQ system, as distinct from physics event data from the detector systems. The interfaces between the DAQ software and the logging system are defined as follows:

- Provide API(s) for DAQ applications to publish structured monitoring data.
- The operational monitoring system shall provide an API in order to be able to output different amounts of information according to a set level at a configurable time interval.
- The operational monitoring system provides an API to subscribe to operational data using a set of criteria.

3.7.1.2.2 Logging

The CCM is responsible for providing a complete solution for capturing, distributing and archiving logs. Logs in this context are debug statements, information, warning and error messages. While debug statements can be unstructured, all higher-level types of messages are structured, carry a well-defined extendable set of information fields and extend from a base class that can be thrown and caught as an exception. The interfaces between the DAQ software and the logging system are defined as follows:

Debugging

- The logging system provides an API to log unstructured messages for debugging purposes.
- The logging system provides tools to dynamically turn on/off or tune the level and type of debug statements which are output.

Structured information

- The logging system provides the base class from which all the messages need to extend as well as helper macros to define specific types. Note: DUNE chose to use the ERS library developed within ATLAS and the base class is the ers::Issue.
- The logging system provides an API in order to be able to output messages according to their severity.
- The logging system provides the tools to configure the implementation of the streams used for each severity.
- The logging system provides an API to subscribe to messages using a set of criteria (e.g. subscribe by message type, application name, host name, severity, etc.).

3.7.2 Specifications

3.7.2.1 Access

The CCM is required to perform authentication and authorization to restrict access to DUNE DAQ resources and actions. The access control system shall provide the following functionality:

- A system to authenticate DUNE users e.g., through FNAL services password.
- Functionality to define user roles and assign (multiple) roles to users e.g., DAQ shifter, detector calibration expert.
- Functionality to authenticate users for these roles. The roles define a set of allowed actions in each CCM subsystem, i.e., control, configuration and monitoring.
- Functionality to propagate access authorization appropriately to each CCM subsystem.

3.7.2.2 Control

The CCM is required to control the DAQ resources to allow DUNE to take data with high uptime. For each of a number of separate partitions, defined by separate DAQ resources in almost all cases, the CCM run control system shall provide configuration and control of a hierarchy of DAQ applications. The run control system shall provide the following functionality:

- An interface for interaction with users, via command line and GUI, with equivalent functionality.
- The interface shall provide functionality to allow the sending of commands, with configuration data to processes, and receiving acknowledgement, to individual processes or to all processes, in a given order.
- The commands for DAQ applications will include a set of per-process defined finite state machine events (typically init, config, start, pause, resume, stop, scrap).
- The system shall provide functionality to allow processes to transition through states in a pre-defined order, requiring the completion of previous process state transitions.
- In addition to state transition commands, the system shall provide expert or operator commands which do not request state transitions perform some operation on the specifically addressed component e.g., setting random trigger rate or toggling data storage.
- The system shall allow the sending of run settings data to processes, using the command interface above, e.g., a new run number when sending the start command.
- Provide a mechanism for accessing a list of available resources for DAQ components, including the mapping between detector readout units (e.g. FELIX cards), processes etc.
- Provide process management, managing the life-cycle of applications, defined as spawning of processes, monitoring of process status, killing and restarting processes for a list of required processes determined through configuration. The system shall also provide process information where necessary e.g., dynamically allocated port information to establish inter-process communication.
- Provide error handling based on aggregation of monitoring information, from a number of systems including operational monitoring, logging, slow control, event data quality monitoring etc., including autonomous actions.
- Archival of per run information in a run database, including the configuration applied, resources allocated etc.

3.7.2.3 Configuration

The CCM is required to provide functionality to prepare, archive, validate and deliver configuration data, including the definition of required partitions. Specifically, the configuration system shall provide the following functionality:

- An interface for interaction with users, via command line and GUI, with equivalent functionality.
- A means by which DAQ application and detector component experts may specify the structure and constraints which describe the relevant configuration information.
- The configuration system shall provide a configuration generator/editor, capable of creating or loading DAQ subsystem configuration data, modifying configuration data, and storing versioned configuration data.
- The configuration system shall provide a mechanism for checking the validity and consistency of configuration data with the DAQ and detector components.

3.7.2.4 Monitoring

3.7.2.4.1 Operational monitoring

The CCM is responsible for providing a complete solution for capturing, distributing and archiving operational monitoring information. Operational monitoring information in this context is defined as information related to the operation of the components of the DAQ system, as distinct from physics event data from the detector systems. Specifically, the operational monitoring system shall provide the following functionality:

- Provide an API for DAQ applications to publish structured monitoring data.
- Provide an implementation for configurable levels of detail in monitoring information.
- Provide an implementation for configurable polling intervals for monitoring information.
- Provide a set of stream implementations for the dispatching of monitoring information, including basic streams to stderr and stdout, as well as network-based streams.
- Provide an archival system and a UI to search for and analyze data, live and post-mortem.
- Provide a system to implement aggregation and analysis functionality.
- Provide alerts and notifications based on defined conditions to the user and to the CCM control subsystem where they may be used to perform autonomous actions.

3.7.2.4.2 Logging

The CCM is responsible for providing a complete solution for capturing, distributing and archiving logs. Logs in this context are defined as debug statements, information, warning and error messages. While debug statements can be unstructured, all higher-level types of messages are structured, carry a well-defined set of information fields and extend from a base class that can be thrown and caught as an exception. The logging system shall provide the following functionality:

- Provide a base class for all structured messages and tools to generate specific messages in a straightforward way.
- Provide APIs for reporting debug text and messages.
- Provide an API to subscribe to messages using a set of criteria.

- Provide a set of stream implementations for the dispatching of debug text and structured messages. Those should include, filtering streams, basic streams to stderr and stdout, as well as network-based streams.
- Provide a mechanism to receive structured messages over the network, for message subscription.
- Provide a message archiving system and a UI to search for and analyze messages, live and post-mortem.

3.8 DQM

3.8.1 Interfaces

3.8.2 Specifications

3.9 Programming Model

If the component contains registers or must be programmed, clearly state whether the registers are write/read or read-only and how the programming is to be accomplished. If some specific software environment or computer system is required to program or operate the component, that should also be described. These points could be covered in a separate sub-section or interspersed with the functional descriptions, depending upon which makes the document more understandable.

4 Development model

5 Testing, Validation and Commissioning

Describe testing procedures that will be used to demonstrate that the fabricated component meets the specifications. This should include radiation testing if radiation tolerance is one of the requirements. It is not mandatory to complete this section prior to the first specification review but it should be completed prior to the PDR. Prior to the FDR, the production testing must be described, and prior to the PRR, commissioning plans must be included.

6 Reliability Matters

6.1 Consequence of Failures

Describe the consequences to the detector of a failure of one unit of this component, e.g. x% of the subdetector channels will be lost. The severity of the consequences will determine the level of reliability required and the level to be validated by QA and QC procedures defined in Sections 6.4 and 6.5.

6.2 Prior Knowledge of Expected Reliability

Based upon industry experience, collaboration experience or personal experience, give an estimate of the reliability of this component.

6.3 Measures Proposed to Insure Reliability of Component and/or System

Include such measures as conservative design techniques (give specific examples), redundancy and possibilities to replace failed part. If failed part could be replaced, estimate the difficulty and time involved for installing replacements.

6.4 Quality Assurance to Validate Reliability of Design and Construction or Manufacturing Techniques

Describe what stress tests will be applied during the development period to validate the reliability of this component. Give a brief outline of any appropriate reliability theory being used. These tests could involve destructive tests. It is not required to complete this section prior to the first specification review but it must be completed prior to the PDR. It is strongly recommended that these plans be reviewed and approved prior to the actual PDR to avoid the possibility of failing the PDR and thus delaying the fabrication or construction of the prototype parts

6.5 Quality Control to Validate Reliability Specifications during Production

Describe what stress tests will be applied during production, possibly on a sampling basis, to validate the reliability of production units. These could likely be destructive tests. Specify the required sampling percentage of production units. It is not required to complete this section prior to the first specification review but it must be completed prior to the FDR. It is strongly recommended that these plans be reviewed and approved prior to the actual FDR to avoid the possibility of failing the FDR and thus delaying the fabrication or construction of the pre-production parts,

References

[1] B. Abi et al. “The DUNE Far Detector Interim Design Report, Volume 3: Dual-Phase Module”. In: (July 2018). arXiv: `1807.10340 [physics.ins-det]`.

[2] B. Abi et al. “The Single-Phase ProtoDUNE Technical Design Report”. In: (2017). arXiv: `1706.07081 [physics.ins-det]`.

[3] C. Mazza et. al. Prentice Hall, 1994. ISBN: 9780131065680. URL: `https://books.google.fr/books?id=0NZQAAAAYAAJ`.

[4] *LBNF/DUNE Requirements and Specifications*. `https://edms.cern.ch/project/CERN-0000198204`.

7 Specification Document Appendices

Any technical detailed information that supports the Specification Document to be used as basis for the SVR should be included in a standalone appendix chapter. It may be useful for example to include technical details, in particular from external inter-dependencies, e.g. GBT and/or lpGBT cases.

The remaining appendices are simply a collection of semi-technical information on a few dedicated LaTeX environments that are being standardised for any TDAQ document.

8 Requirements and their classification, Specification rules

The requirements should be organized according to the classification described in [3].

8.1 Requirements

Mandatory ***Requirements*** are sentences that contain the word ’shall’. These requirements must be followed. Two forms are defined for the requirement environment: The standard (or long) form has 6 arguments, which have to follow a specific order:

1. A generic key specifying the requirement type, e.g.: GEN, CONV, MAIN, ENV, DEP, INTFC
2. Title capturing a short description of the requirement
3. Requirement type: essential/non-essential
4. Priority: 1…3, Suspended
5. Stability: Stable (e.g. frozen), Unstable (subject still to evolution)
6. Reference: either Person, Group, System, Sub-system owning the requirement

Example:

```
\begin{requirement}[MAIN]{Roger Federer Requirement}{Essential}{1}{Status}{R. Federer}
\label{req:long_roger_requirement}
 Roger shall win 10 Wimbledon titles before retiring
\end{requirement}
```

Requirement 8.1 [REQ-MAIN]: *Roger Federer Requirement*

Essential1StatusR. Federer Roger shall win 10 Wimbledon titles

Optionally, short forms are allowed, but the second argument, i.e. the first between curly brackets that specifies the requirement description, is mandatory. In this case a generic identifying key is assigned to the requirement:

Example:

```
\begin{requirement}{Roger Federer short requirement}
\label{req:short_roger_requirement}
 Roger shall win 10 Wimbledons
\end{requirement}
```

Requirement 8.2 [REQ-8.2]: *Roger Federer short requirement*

Roger shall win 10 Wimbledons

8.2 Recommendations

Recommendations are requirements that use the word 'should'. If not followed a justification is needed. A dedicated requirement has been implemented in the TDAQGenDoc class.

Example:

```
\begin{recommendation}{Roger Federer recommendation}
\label{rec:roger_backhand}
 Roger should always counterpunch with his backhand
\end{recommendation}
```

Recommendation 8.1 [REC-8.1]: *Roger Federer backhands*

Roger should play more top-spin backhands

Recommendation 8.2 [REC-SAV-GENERAL]: *Roger Federer serve and volley*

Roger should play serve and volley more often

8.3 Guidelines

Guidelines are requirements containing the word 'may'. If they are not followed, no justification is required. An environment specifying a guideline has been implemented in the document class.

Example:

```
\begin{guideline}{Roger Federer guideline}
\label{gui:roger_guideline}
 Roger may want to go to net more often
\end{guideline}
```

Guideline 8.1 [GUI-8.1]: *Roger Federer guideline on net approach*

Roger may want to go to net more often

Guideline 8.2 [GUI-CLAY-ROLGAR]: *Roger Federer guideline on clay season*

Roger may want to play once more Roland Garros

8.4 Remarks

Remarks are generic statements that you may want to highlight wrt. the regular text.

Example:

```
\begin{remark}
\label{rem:roger_remark}
 Roger has won so far 8 Wimbledon titles
\end{remark}
```

Remark:

Roger has won so far 8 Wimbledon titles

8.5 Definitions

Definition environment are used

Example:

```
\begin{definition}{Roger Federer}
\label{def:roger_definition}
 Roger is a Swiss tennis player, born in Basel on 12/8/2018, and considered the tennis GOAT.
\end{definition}
```

Definition 8.5.1 [DEF-8.5.1]: *Roger Federer*

Roger is a Swiss tennis player, born in Basel on 12/8/2018, and considered the tennis GOAT.

8.6 Specification Rules

Specification rules are derived technical requirements to be satisfied by a design, product, service and/or material. It is implemented as a separate latex environment, with a format similar to the requirement environment.

Example:

```
\begin{specrule}{Roger forehand specification}
\label{spec:roger_forehand}
Roger forehand bases on an Eastern grip that allows him to hit both with
topspin and flat. Its preparation consists of a "body" turn, i.e. racket,
shoulder and lower body turning together initially to create the coil and store
the energy for the swing. The swing is realised through the rotation of his upper body.
During the backswing of the arm remains aligned
to the body's trunk, followed by a whip-like movement of the arm and  of the wrist.
The finish utilises the classic "over the shoulder" movement.
\end{specrule}
```

Specification 8.1 [SPEC-8.1]: *Roger forehand specification*

Roger forehand bases on an Eastern grip that allows him to hit both with topspin and flat. Its preparation consists of a "body" turn, i.e. racket, shoulder and lower body turning together initially to create the coil and store the energy for the swing. The swing is realised through the rotation of his upper body. During the backswing of the arm remains aligned to the body's trunk, followed by a whip-like movement of the arm and of the wrist. The finish utilises the classic "over the shoulder" movement.

Optionally, a second argument can be specified to provide an identifying key to the specification rule:

```
\begin{specrule}[SPEC-TECH-FORE]{Roger forehand specification}
\label{spec:roger_forehand_ext}
Roger forehand bases on an Eastern grip that allows him to hit both
with topspin and flat. Its preparation consists of a "body" turn, i.e. racket,
shoulder and lower body turning together initially to create the coil and
store the energy for the swing. The swing is realised through the rotation
of his upper body. During the backswing of the arm remains aligned
to the body's trunk, followed by a whip-like movement of the arm and  of the wrist.
The finish utilises the classic "over the shoulder" movement.
\end{specrule}
```

Specification 8.2 [SPEC-TECH-FORE]: *Roger forehand specification*

Roger forehand bases on an Eastern grip that allows him to hit both with topspin and flat. Its preparation consists of a "body" turn, i.e. racket, shoulder and lower body turning together initially to create the coil and store the energy for the swing. The swing is realised through the rotation of his upper body. During the backswing of the arm remains aligned to the body's trunk, followed by a whip-like movement of the arm and of the wrist. The finish utilises the classic "over the shoulder" movement.

9 Tables

The TDAQGenDoc class supports several possible table packages. In addition, a standard table template to be used for the formal TDAQ through the *tdaqTable* environment. The environment is defined with 6 mandatory arguments (remember in latex the 2nd argument is indicated between [ ]) and a 7th optional argument:

1. Positional argument, e.g. `[htbp]`
2. Size of the table columns
3. Caption
4. Table's label
5. Header row
6. Table's body
7. Table footnotes

Table 9.1: ATLAS HL-LHC Upgrade plans and reference to the detector system TDR.

System	CDS Reference
ITk Pixel	CERN-LHCC-2017-021
ITk Strip	CERN-LHCC-2017-005
LAr Calorimeter	CERN-LHCC-2017-018
Tile Calorimeter	CERN-LHCC-2017-019
Muon Spectrometer	CERN-LHCC-2017-017
TDAQ	CERN-LHCC-2017-020

Example:

```
\begin{tdaqTable}[htbp]{L{0.68\textwidth}R{0.22\textwidth}}
{\ATLAS \gls{HL-LHC} Upgrade plans and reference to the detector system \glspl{TDR}.}
{tb:atlas-upgrade-plans}
{System & CDS Reference}
{%
ITk  Pixel    &  \href{https://cds.cern.ch/record/2285585}{CERN-LHCC-2017-021} \\
ITk Strip     &   \href{https://cds.cern.ch/record/2257755}{CERN-LHCC-2017-005}
\\ \midrule
LAr Calorimeter &\href{https://cds.cern.ch/record/2285582}{CERN-LHCC-2017-018}\\
Tile Calorimeter & \href{https://cds.cern.ch/record/2285583}{CERN-LHCC-2017-019}
\\  \midrule
Muon Spectrometer &   \href{https://cds.cern.ch/record/2285580}{CERN-LHCC-2017-017}
 \\ \midrule
TDAQ          &
\href{https://cds.cern.ch/record/2285584/files/ATLAS-TDR-029.pdf}{CERN-LHCC-2017-020}
\\ \bottomrule
}
\end{tdaqTable}
```