Outline

- DCS Overview and Update
  - ALFRED
  - WinCC OA Control Application
- QC Overview and Update
- Future Plans
DCS Layout: Overview

- DCS = Detector Control System
- Encapsulates all necessary automatic (e.g. software interlocks) and manual (e.g. configuration, masking/unmasking, etc.) detector control tasks
- Multilayered system, with different layers having both different functions and different levels of specialization

Figure from Peter Chochula, ALICE
DCS Layout: $O^2$ Layer

- Online-Offline Software ($O^2$), common to all subdetectors, handles operation of First Level Processor (FLP)
  - $FLP$ contains one or more ALF servers
  - ALF servers handle basic communication with Common Readout Unit (CRU)
  - CRU interfaces with many Readout Units (RU), which communicate directly with ALPIDE chips and power boards
- Communication protocol can be either SCA or SWT (ITS uses SWT, most other subdetectors use SCA)
DCS Layout: FRED Layer

- Subdetector customization possible via FRED server
  - Translates high-level commands to sequences of SCA/SWT words, sends to ALF server
  - Manages concurrent access via RPC queue
- Configuration and monitoring sequences implemented in FRED
- Communication via DIM
- ALF+FRED=ALFRED

Figure from Peter Chochula, ALICE
DCS Layout: UI Layer

- Provides graphical interface to ALFRED
- Implements finite-state machine
- Performs monitoring tasks and database connection
- Contains all necessary detector control functions
- Can be either WinCC OA SCADA system or standalone C++ application (or both)

Figure from Peter Chochula, ALICE
WinCC OA Control Application: Overview

- WinCC Open Architecture (OA): object-oriented SCADA system used for industrial control applications
- WinCC OA application provides graphical interface to ALFRED system
  - Sends high-level task command sequences to ALFRED (e.g. “Configure stave L0_03”)
- Also controls the other ITS components, for example:
  - Configures and controls power boards/other electronics
  - Manages access control

Figure from Peter Chochula, ALICE
WinCC OA Control Application: Current Status

- For now, need something that can be used quickly for IB commissioning
  - May not necessarily reflect final application structure
- Uses only low-level read/write functions in ALFRED, while keeping most of the logic in WinCC
  - Limited to controlling chips and readout units at the moment
- Focus on simplicity and short development time
- Panel architecture may also be usable in final iteration of control application
WinCC OA Control Application: Panel Topology

Graphical ITS Display

- Select Stave from List
  - Stave List
    - OK
      - Stave Control Window

- (click on stave)

Access Config DB

Config Database Editor

Config Database

Autoconfigure

- Manual Configure
  - Manual Config Options
    - Broadcast Masking
      - Pixel Matrix
        - Pixel Matrix Display
      - Mask Chip
        - Pixel Matrix Display
      - Direct Chip Read/Write
    - Direct Chip R/W Interface
  - ALFRED DIM Datapoints

WinCC OA Control Application: Panel Topology

- Basic control unit: the ITS stave
- Two ways to select stave:
  - Graphically, based on geometric location
  - From list, based on established naming convention
WinCC OA Control Application: Panel Topology

- Once stave is selected, opens stave control panel
- “Autoconfigure”: access configuration database (file database for now) and apply settings to this stave
- “Manual Configure”: open manual configuration menu

(database query)
WinCC OA Control Application: Panel Topology

- “Mask/Unmask Chips”: mask/unmask entire chip
- “Pixel Matrix”: Access pixel matrix commands for individual chip
  - Commands can be applied to both a single row/col/pixel or a list
  - Matrix display (not yet implemented)
WinCC OA Control Application: Panel Topology

- “Direct Chip Read/Write”:
  - Manually write opcodes to command register
  - Manually write/read data from other registers
  - Can specify one chip or broadcast to all chips in stave
WinCC OA Control Application: Panel Topology

- “Access Configuration Database”: Read from and write to configuration database that sets parameters for “Autoconfigure”
- Currently just a placeholder panel, until final structure is decided
QC Framework Overview

Figure from Barthelemy von Haller, ALICE
QC Usage

- User creates QC Task
  - Takes some event data as input (currently using raw)
  - Generates desired monitoring objects (e.g. histograms) as output
- User creates QC Checker (optional)
  - Takes monitoring objects as input
  - Passes monitoring objects and produces quality metric as output
- User creates JSON file linking web API object to database entries
- User accesses web API for QC monitoring
QC Current Implementation

Target date is up to the ITS developers

Figure from Barthelemy von Haller, ALICE
QC Web API: Current Status

- Being developed by Zhaozhong Shi and others
- QC objects stored in CCDB instance here: http://ccdb-test.cern.ch:8080/browse/ITS QcTask/
- Customizable layouts, QC objects, refresh period

Figure from Zhaozhong Shi, MIT
Future Plans

● QC:
  ○ *Build up standard library of ITS QC Tasks and QC Checkers*
  ○ *Test new QC features (DPL features, correlation, trending, machine learning, etc.) as they are implemented*
  ○ *Fully implement and benchmark CCDB instance for storing monitoring objects*

● DCS:
  ○ *ALFRED:*
    ■ Continue to develop and use high-level functionality within FRED
    ■ Build up a standard library of SWT sequences (where necessary)/C++ functions (if possible) for ITS configuration and operation
      ■ *Porting existing Python libraries to C++*
      ■ *Includes all configuration and monitoring features, as well as the logic for partitions*
Future Plans (cont.)

- DCS:
  - WinCC OA Control Application:
    - Implement “Autoconfigure” within application for commissioning purposes (in progress, but nearly finished)
    - Develop interfaces for chip, RU, powerboard and monitoring
    - Design configuration database and implement a corresponding interface
    - Implement finite state machine
    - Interlink service and configuration projects
    - Adapt current components for operations version
Summary

- ALFRED: Allows high-level commands to be translated into communications with readout chip and power board
  - *Current status*: High-level functionality implemented, stave control Python scripts being ported to C++ for use with MAPI.
- WinCC OA: Acts as graphical interface for ALFRED and database for monitoring and possibly configuration
  - *Current status*: Commissioning-level control application designed, full feature set being implemented.
- QC: Provides workflow for direct monitoring of detector data via web API.
  - *Current status*: Test workflow implemented, feature list being expanded.
- Looking forward to applying this expertise to MVTX!
BACKUP
DCS Usage

- Various ways of generating SCA/SWT sequence, vary in speed, safety and easiness
- Usage of ALFRED prevents concurrent-access conflicts (is “safe”)
- In principle, custom client can directly access ALF (“unsafe”)