User Stories

Collection of Inputs / Collation of Requirements

Simon Spannagel, DESY
2nd EDDA Meeting
11/10/2023

Recap: Build a flexible SCADA software system

- **Useful to control single laboratory setup** (e.g. radioactive source measurement)
  User story: Start measurement in the lab, return to office. Open up monitoring tool, observe data trickling in, temperature being stable. Check logs for anomalies. Go to meeting. Receive phone notification about finished measurement, abort meeting.
- **Possibility to integrate multiple setups** (Detector DAQ, TCT laser control)
  User story: Write common configuration file detailing detector parameters and laser driver/stage configuration. Start measurement.
- **Lab supervision mode** (multiple setups monitored but control not ceded)
  User story: Central monitoring of multiple lab setups, but not necessarily controlled via common Control Center but from individual lab workstations
- **Synchronized operations** (test beam environment, coordinated start/stop, central control)
  User Story: Run multiple systems in parallel with a central control; automated parameter scans, resilience to failures like graceful restart
- **Scalability for experiments** (many detectors, multiple data endpoints & monitors)

Derived Requirements – System Architecture

- Excellent documentation of architecture & APIs
- Simple / straight forward interface with few methods to implement
- Flexible build system to only build parts of the framework

Derived Requirements – Finite State Machine / Control

- Distributed state machine capable of fault-tolerant behavior
- Possibility to define FSM response to incidents per-satellite
- Possibility to fully recover to “non-degraded” state after rejoining of satellites
- Possibility to influence distributed state of constellation from individual satellites (request start, stop)
- State machine error state needs to be safe, satellites e.g. power down
- Either standalone mode w/o control center possible or executable that starts satellite with “bundled” control center (transparent)
- Ensure re-configuration of devices & DAQ start/stop is quick enough to be usable in fast scan loop
- Possibility to define dependency graph for startup / down (systemd-like?)
- Centrally managed access control to state machine, possibly: UI needs to request control, all others fall back into monitoring-only

Derived Requirements – Data Handling

- Architecture that separates control code from data decoding
- Possibility to collate data from multiple satellites in single storage node,
  Possibility to partition / split data streams on storage nodes (multiple files / shards)
- Possibility for synchronizing data based e.g. on trigger ID information
- Satellite that can dispatch stored data to streams
- Possibility for user-defined monitoring data
- Monitoring data: think about time granularity, integrating & differential variables, have possibility to communicate type of data to monitor
- Monitoring endpoint that can collate & plot variables flexibly
- Possibility to log to searchable database
- Possibility to store statistics / monitoring data in database
- Allow satellites to provide metadata for storage (data stream, statistics)

Derived Requirements – UI / Monitoring

- CLI interface to connect to {satellite|control center} for control, no GUI of CC required
- Flexible multi-run / scanning feature that is capable of reconfiguring the entire constellation
- Logging interfaces / alarms for messengers (Mattermost, Signal, Mail …)
- provide possibility to expose current settings to GUI or monitoring tool (re-distribute settings as statistics data?)
- Possibility of seamlessly remove debugging / switch to production mode (flexible logging, debug statements without (with minimal) runtime penalty)
- Provide flexible interface for scanning feature (including ranges with fixed binning, variable bin sizes, list of absolute values)
- Possibility to remotely connect to constellation for control purposes

Derived Requirements – Other Features

- Set of standard satellites for commonly used hardware such as EPICS-controlled crates
- Think of resilient solutions to (partial) network outages
- Possibility for satellites with secondary (GSM) connection for alerts?

User Stories – I

Adrian Herkert

Low Entry Threshold

Test beam users (who are not software-developers) want to add a DUT to an existing 'Constellation' (e.g. TLU plus beam telescope). I point them to a document including step-by-step instructions on how to achieve that. Within a day, they manage to install Constellation, link it to their own libraries, and execute all the steps in the instructions. They end up with the bare minimum of a 'Satellite' where they can then start plugging in their required functionality.

Derived Requirements

- Excellent documentation of architecture & APIs
- Simple / straight forward interface with few methods to implement
- Flexible build system
- (multiple language bindings)

User Stories – II

Adrian Herkert

Standard Data Format / Helpful, Intuitive Build Options

Constellation gives you the option to use a standardized compact data format for writing measurement data to disk. A student working off-site on test beam data analysis wants to use Constellation to read back data in said format. It would be a lot o unnecessary work to install all DAQ software libraries that were required to produce the data. They can avoid this by installing Constellation with the build option "Build only analysis-relevant modules".

Derived Requirements

- Flexible build system
- (multiple language bindings)
- Architecture that separates satellite code (SCADA) from data decoding

User Stories – III

Adrian Herkert

Flexible Online Monitor

I would like to add a specific type of plot to the available online monitor that hasn't been implemented yet. Constellation offers me a way of doing that.

Derived Requirements

- Possibility for user-defined statistics data
- Think about time granularity, integrating & differential variables, have possibility to communicate type of data to monitor
- Monitoring endpoint that can collate & plot variables flexibly
- Should this be based on “statistics” channel entirely? No decoding, no touching of actual data stream, but needs separate data generation by satellite

User Stories – IV

Bernhard Pilsl

Lab Characterization

I'm characterizing a detector in the lab. The baseline DAQ is the Caribou System (Yocto / Poky Linux on SoC). I want to perform automated multidimensional scans (chip configs (I2C / SPI), bias voltage, etc.). A CAEN-crate does the biasing (controlled via EPICS interface). I want notifications for progress milestones and (especially) for errors on my phone.

Derived Requirements

- Flexible multi-run / scanning feature that is capable of reconfiguring the entire constellation
- Set of standard satellites for commonly used hardware such as EPICS-controlled crates
- Logging interfaces / alarms for messengers (Mattermost, Signal, Mail …)

User Stories – V

Bernhard Pilsl

Testbeam Characterization

Again, detector characterization, but this time at a test beam facility (e.g., DESY 😉). I want something EUDAQ-like but better / more flexible / more stable. So, I want to integrate my sensor into the DAQ used at the facility as easily as possible.

Furthermore, I want a compact data format that can quickly be converted to other formats (e.g. ROOT) for easy debugging.

Derived Requirements

- Excellent documentation of architecture & APIs
- Simple / straight forward interface with few methods to implement
- Architecture that separates satellite code (SCADA) from data decoding
- (providing multiple data formats and converters between them)

User Stories – VI

Bernhard Pilsl

Single-setup Use Case

I want a system as flexible as possible so it could replace e.g. Peary and EUDAQ in one go. For this to work, a standalone mode of the satellite (no connection to a control center) e.g. a CLI interface would be handy for developing and debugging.

Derived requirements

- Either standalone mode w/o control center possible or executable that starts satellite with “bundled” control center (transparent)
- CLI interface to connect to {satellite|control center} for control, no GUI of CC required

User Stories – VII

Uwe Kraemer

Multi nested variable scans with an easy look at current settings.

I have a laser with which I wish to perform in-pixel measurements. The laser position is controlled via software and so are a PS for biasing and the laser intensity. I wish for the system to now perform a nested scan through all these parameters. I would also like to not only see how far along it is relative to the full run but also at a glance which of the variable parameters it is currently taking data with. This then allows me to see in the online monitoring whether it looks fine or not since not everything that is not normal is an error.

Derived Requirements

- Similar to User Story IV
- In addition, provide possibility to expose current settings to GUI or monitoring tool (re-distribute settings as statistics data?)

User Stories – VIII

Uwe Kraemer

Combined changes of parameters in a scan

Laser measurements of a multi pixel matrix in which only a single pixel responds each time. When I move the laser from one pixel to another, I wish to, at the same time tell the system to also send a command to whatever device that enables the pixel that lies beneath the new laser position to enable it instead.

Derived Requirements

- Similar to User Story IV
- Ensure re-configuration of devices & DAQ start/stop is quick enough to be usable with shot-move-shot-move-… pattern

User Stories – IX

Uwe Kraemer

Unequal changes of different parameters in a scan.

I investigate the crossing point between two analog pixels with a laser. I want to be able to move towards the edge in small steps for example from 0 to 30 in 1 micrometer steps. Only once it reaches the value 15 micron I want to change which pixel I read out and then continue onwards to the end. Ideally I would be able to define multiple such crossing points.

Unequal changes of a single parameter during a scan

Performing a measurement of the full pixel I am mostly interested in the edges instead of the core. As such I would like to tell my scan to have different granularity at different points to focus on those regions of interest. I know my edge starts at 10 micron so I want it to move by 5 micron before and 1 micron afterwards. Or even better is that I can assign a formula which defines the stepping that the system follows where for example the step size follows a 1/x dependence where x is the step number.

Derived Requirements:

- Provide flexible interface for scanning feature (including ranges with fixed binning, variable bin sizes, list of absolute values)
- (possibility for “scripts” using if/else of statistics data?)

User Stories – X

Hanno Perrey

Long-running experiments: Quick prototyping

During the initial development stages, Constellation provides detailed logging output and provides debugging hooks to quickly identify issues and fix them. This helps to deal with issues that are hard to reproduce or only happen after longer runs.

Derived Requirements

- Similar to User Story I
- Possibility of seamlessly remove debugging / switch to production mode (flexible logging, debug statements without (with minimal) runtime penalty)

User Stories – XI

Hanno Perrey

Long-running experiments: Recovery processes

Sometimes, one has to live with quirky hardware and rare but expected faults. By providing hooks for fault conditions, Constellation can recover from specific issues by executing a pre-defined protokoll to e.g. restart a power supply. If a user is present, they have a chance to intervene, before the recovery process is started. Such incidents are logged, of course. Any running acquisition is continued in a sensible fashion defined by the user beforehand.

Derived Requirements

- Distributed state machine that is capable of fault-tolerant behavior
- Possibility to define control center response to incidents per-satellite (“satellite importance”)

User Stories – XII

Hanno Perrey

Long-running experiments: Long Up Times

Constellation can be easily run for weeks-long or even months-long acquisitions. Logs provide insights into any issues that were encountered recently but would not fill the disk. Minor failures (such as a temperature sensor not providing data) are noticed but can be ignored. Once the sensor is back up, it is re-integrated into the data stream.

Derived Requirements

- Extends upon User Story XI
- Possibility to partition / split data streams on storage nodes
- Possibility to fully recover to “non-degraded” state after rejoining of satellites

User Stories – XIII

Hanno Perrey

Long-running experiments: Automatised acquisition start

It is possible to Start/Stop the acquisition externally, e.g. via a script, to automatize processes or to configure external conditions. Maybe one only wants to start the acquisition once a temperature sensor sends a certain value or some other condition is met.

Derived Requirements

- Similar to User Story VI
- Possibility to influence distributed state of constellation from individual satellites

User Stories – XIV

Annika Vauth

Lab setup

I'm using a laser, moveable stage, power supplies, oscilloscope to do measurements with some sample sensors. The laser is currently steered separately from a windows PC using the manufacturers software, but everything else can be controlled via SCPI and GPIB commands. One example measurement is moving the stage so that the laser spot is close to an opening in the passivation layer of the sensor, then scanning it in micrometer-steps across the opening and recording oscilloscope data at each step. Repeat at different bias voltage. Possible online monitoring: retrieve one of the scope measurement parameters, e.g. area under the signal, and plot it vs stage position. Monitoring the current of the power supply would also be welcome. Possible error handling: the stage controller sometimes crashes. Catch timeout as exception and switch off bias voltage if it happens.

(*) On the scanning: an option for more "complicated" parameter ranges would be good. E.g. to scan three passivation windows, but save time by not measuring at the positions in between. Bonus feature: randomize the order ;-)

Derived Requirements

- Similar to User Story IX
- State machine error state needs to be safe, satellites e.g. power down

User Stories – XV

Lene Kristian Bryngemark

System integration/event building

Me and my colleagues are testing several detector prototypes together (in beam, but might just as well be cosmics), which have advanced readout electronics prototypes too that take care of writing a complete event output (headers, digitized signal etc) from each detector. We have several triggers and beam instrumentation in the setup, but no system for distributing a clock when we come to the testbeam facility. We want to be able to combine the information measured in beam instrumentation (e.g. beam trajectory) with all our detector measurements by having synced up output built for each triggered event.

Derived Requirements

- Possibility to collate data from multiple satellites in single storage node
- Possibility for synchronizing data based e.g. on trigger ID information

User Stories – XVI

Lene Kristian Bryngemark

Testbeam prep work

When preparing for testbeam datataking, I want to exercise the DAQ machinery to make sure that I can read data from all input sources and prepare decoding/reformatting code that translates bits in a stream to signal on channels in a triggered event, in some format I use for my data analysis software. For this I want to feed expected input source test data (with a format I provide, for the detectors I will bring) to a system emulator.

Derived Requirements

- Satellite that can dispatch stored data to streams

User Stories – XVII

Lene Kristian Bryngemark

Remote shifter

I am taking shifts remotely for the testbeam that my colleagues have set up. I need secure access over network and some method for authentication. I need a remote interface with timely response and stability wrt connection glitches.

Derived Requirements

- Similar to User Story IV
- Possibility to remotely connect to constellation for control purposes

User Stories – XVIII

Lene Kristian Bryngemark

Slow control and conditions logging

I have a detector (or source) mounted on a motorized stage that I want to move for beam scans. I want to set the position as part of run control and log it (at least once per run) in a run conditions log or database that clearly associates these metadata to the detector data taken. I also want to log environmental data such as temperature, humidity, pressure, ... there. While all this could be done by a shifter, that is error prone and requires manual intervention, and I much prefer pulling already digitally available info directly and automatically.

Derived Requirements

- Possibility to log to searchable database
- Possibility to store statistics data in database
- Allow satellites to provide metadata for storage (data stream, statistics)

User Stories – XIX

Lene Kristian Bryngemark

Alert System / Setup Supervision

An electrically cooled germanium detector is ON and it must be running constantly. If the power is off for more than 15 minutes or so, the temperature goes "too high" and it is not recommended to cool it again. A full cooling cycle is needed. We had those detectors on UPS with batteries capable of keeping the systems for 30 minutes or so and were set up to send e-mails when alerts and notifications occur. The issue? well a power failure also takes our the network routers and we never got the notification so a cooling cycle was needed =(

Derived Requirements

- Think of resilient solutions to (partial) network outages
- Possibility for satellites with secondary (GSM) connection for alerts?

User Stories – XX

Finn Feindt

Defined Startup/Rampdown Order

Define ramp down procedures for devices. Define device order for state transitions. Abort save data format like in EUDAQ. Automated start extremely interesting

Derived Requirements

- Possibility to define dependency graph for startup / down (possibly look at systemd approach)

User Stories – XXI

Annika Vauth

Multi-UI & Control Requests

When controlling from remote, we need to make sure that e.g. local in-lab UI has higher priority / we need to come up with a possibility of seizing control from Uis when another UI joins and requests control access.

Derived Requirements

- Centrally managed access control to state machine
- Possibly: UI in monitoring mode, request control, all others fall back into monitoring mode
- (having little chat to discuss?)


DESY.