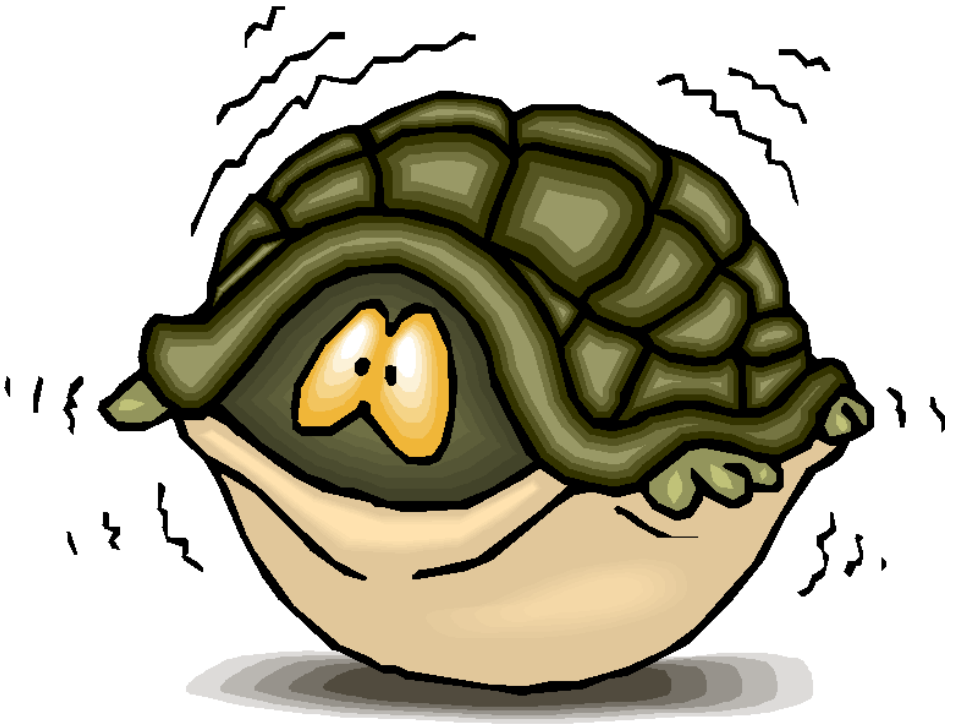
Slow/Run/Whatever control

Mikihiko Nakao (KEK-IPNS)

mikihiko.nakao@kek.jp

January 18, 2012



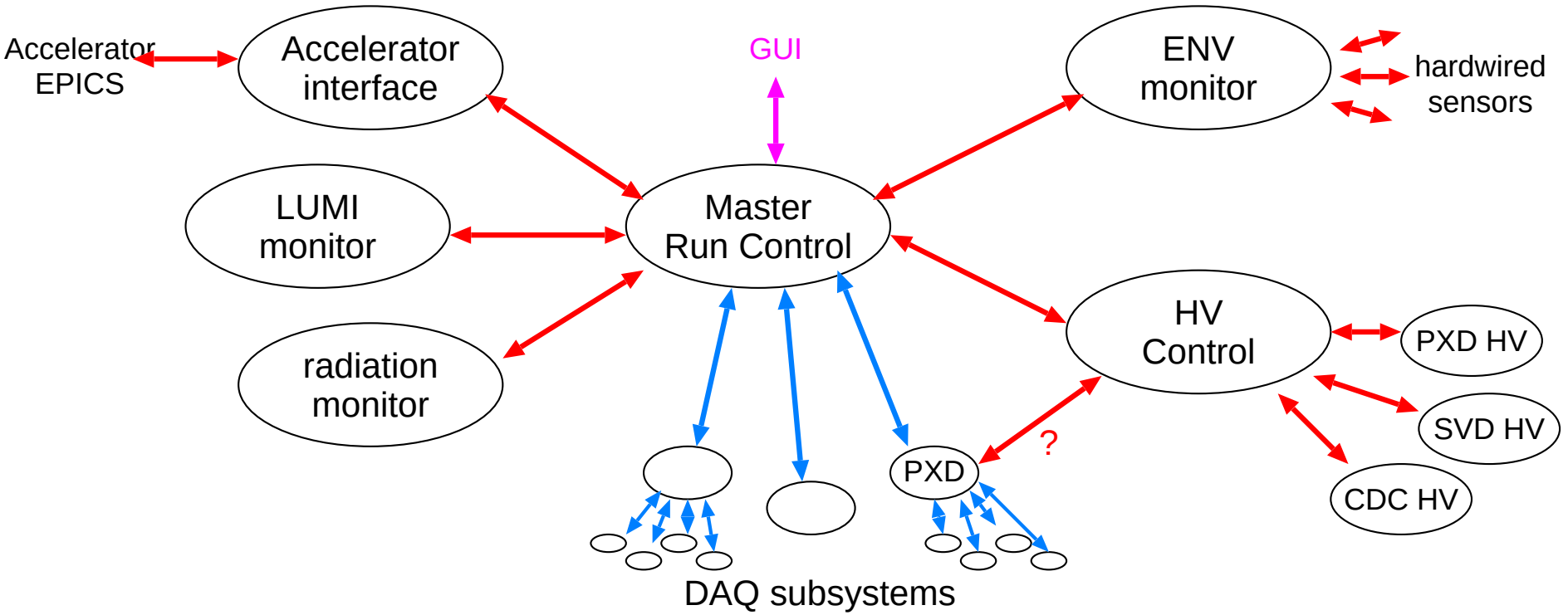
Nothing has been done on my side because of other works on Belle2link and timing systems.

Requirements

inspired by a proposal fro PXD-SC group, with my rearrangements. . .

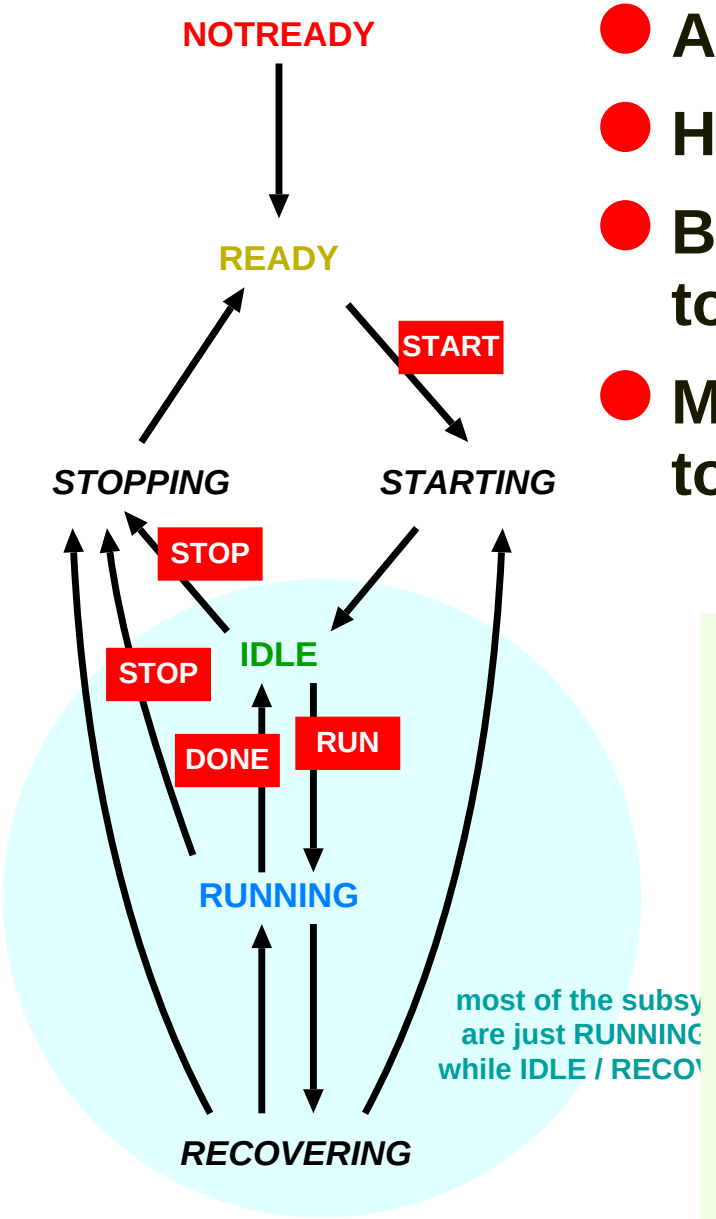
1. **Usability** — design should be made in such a way that a even a theory student can handle
   (a) **Unification** — unified program wherever possible, if not "unified" interface and look & feel as much as possible
   (b) **Abstraction** — classify subsystems into several models, such as DAQ subsystem, HV subsystem, . . .
2. **Accessibility** — control & monitoring, security, realtimeness
   7-day-24-hour fix is needed, even over a very slow network or phone call in Japanese-/Chinese-/Korean-/Indian-/xxx-English communication
3. **Availability** — should not stop working at any time, except for scheduled shutdown time
   (a) **Simpleness** — less layers, less unnecessary functions, . . .
   (b) **Fault tolerance** — redundancy if needed and possible

Control tree

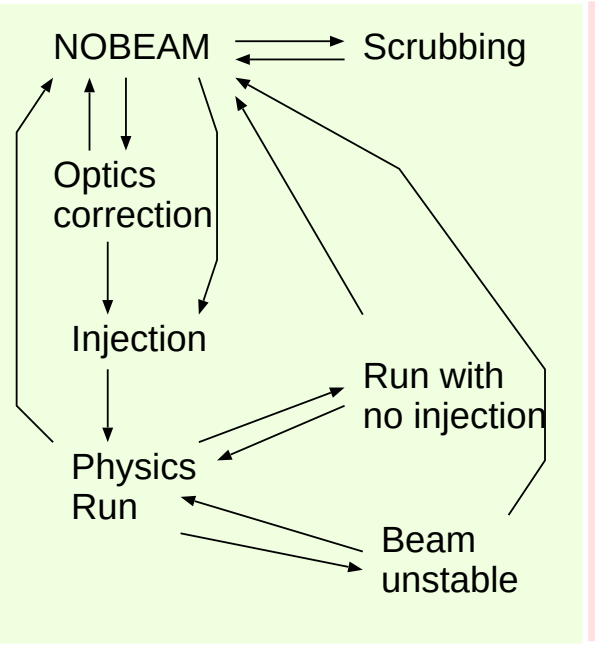


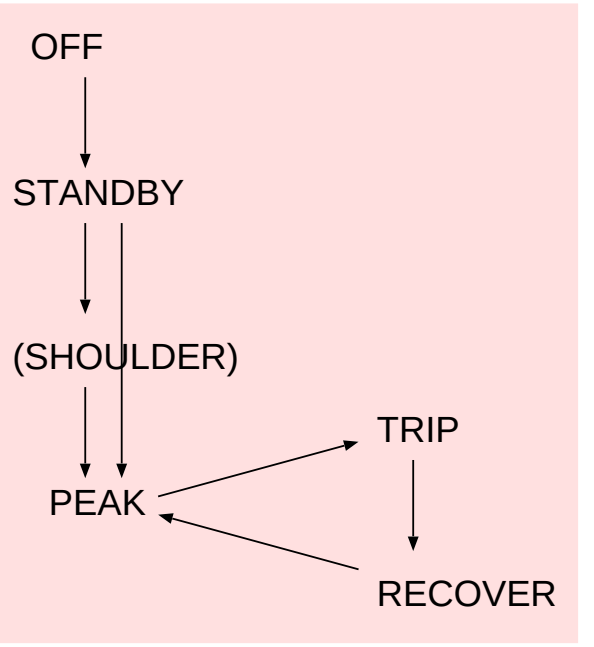
- **All information centralized**
- **Common interface to each detector subsystem**
  - HV and DAQ are separately handled
  - PXD related items were far complicated than others, and tended to be handled as special case (same as Belle I SVD)
- **These top-level subsystems fit well with NSM framework**

State model



- Accelerator has its own state transition
- HV system has its own state transition
- Both state transitions take time and not to be mixed up with DAQ states
- Master control takes them as interlocks to allow / block its own transitions





Network, security, and remote access

- **Security is getting more and more serious issue**
  - fbdaq is a flat network with no security protection
  - Entering point to fbdaq should be minimized (only bdaq and a few more)
  - bdaq account only for limited people, shifter has to stay in the control room
- **Exporting data out from fbdaq is safer, than controling system in fbdaq from outside**
  - Copy of the monitoring terminal (with no control), and/or Web and database based monitoring data export
  - All control only through SSH connections (I don't think it's a good idea to add our own security layer)
  - CUI (text-based) real-time monitors as much as possible

backup slides
(material from PXD-SC group, and PXD-DAQ WS slides)

General Considerations for the Slow Control System of the Pixel Vertex Detector ("PXD") in the Belle II Experiment

A. Knoll, C. Kiesling, J.Schieck

(Dec. 22, 2011)

As was discussed during the B2GM Meeting at KEK in November, we shortly summarize here the most important ingredients and characteristics of a Slow Control system (”SC”) for the PXD. Such guidelines would most likely also apply to a more general system, controlling the Belle II experiment.

We have identified the following core characteristics of the PXD SC:

1. **Unified access:**
   Access to the individual hardware components (”subsystems”) should be implemented with identical metaphors to provide a uniform ”look-and-feel” to support ”intuitive” operation of these components without frequent interventions of experts.

2. **Abstraction:**
   In order to enable direct access and powerful programming of the individual subsystems, methods of extensive abstraction from concrete hardware should be developed. To this end, common features of the subsystems regarding their functionality should be identified and grouped together in a meaningful way (such as ”currents”, ”voltages”, ”temperatures” etc.)

3. **Virtuality:**
   In informatics, it is common use to present to the user a set of systems or devices even if they do not exist as an (undivided) entity in the real world. This concept should be realized also here, e.g. in the form of a ”distributed virtual control room”, with which the steering and control can be executed in the same way as with a classical, physically existing control room.

4. **Connectivity:**
   The entire set of functions of the SC, which do not need direct physical intervention by a human operator, should be accessible in a safe (encoded) way, and only for authorized users, over the world-wide data networks. With such a scheme ”remote monitoring” as well as ”remote control” will be possible in real time, in principle from each point on the globe. This will become particularly important when the entire detector (or essential parts of it) will be operated by a set of expert teams distributed geographically over different countries and even continents. Here, stringent requirements to the real time capability will have to be imposed in order to guarantee safe operation in the presence of communication disturbances and even interruptions, which necessitates the use of appropriate protocols. The goal should be to create a combination of autonomous operation of the devices together with complete ”tele-presence” and ”tele-action” of the operators.

5. **Fault Tolerance:**
   In large complex experimental facilities, such as the PXD system or the Belle II detector in general, there is a certain probability of failure of some component or subsystem in a randomly given time interval. In as much as these components are crucial for the operation (for example power supplies, cooling systems or communication technology), redundancy of such components should be introduced by construction in order to guarantee continuous operation even in case of failure. For this, a sophisticated system of error recognition and error recovery is needed. Also, care has to be taken that faulty components can be serviced or exchanged and re-integrated during continuous operation of the experiment (”hot swap”).

6. **Plug-and-Play:**
   The subsystems of Belle II consist of a multitude of components and devices which are intimately networked amongst each other and the general run-control (e.g. NSM). During the lifetime of the experiment these components will have to be exchanged several times or replaced by upgraded versions. A powerful plug-and-play mechanism would enable such exchanges and upgrades with a minimal effort and down-time.

7. **Serviceability:**
   The design of the entire SC should follow the principle of complete transparency, which not only means that it should be documented well, but that also all properties and parameters should be maintained at an up-to-date level, so that a ”live documentation” emerges. Such measures are essential ingredients and will inevitably lead to effortless maintainability and serviceability.

We understand that all these key characteristics of the PXD SC will not be easy to implement in a first realization within the next two or three years. However, we believe it is important to follow these rules from the beginning in the specifications of the PXD SC and to keep in mind these aspects in all further design decisions during the development of the project.

Run control and NSM

- **Run control for Belle II is currently taken care by me, and it is crucial that KEK is responsible for smooth data taking**
  - Run control requires complex sequencing of multiple inhomogenious systems, and will grow
  - The other slow control systems are a lot simpler systems until PXD brought in their CO2 cooling and power supply systems.
- **Our plan has been to use NSM for inter-process(er) communication as we have legacy** (codes and experiences) **to run it on COPPER, reconstruction farm, etc**
- **NSM is only for IPC, and all the local sequencing is written by a C++ code**
- **I'm happy to see what is done at LHC and learn from it, and I'd like to choose what minimizes my load**

PVSS

- **Clara showed us how to generate a toy example**
  - Define a data point and actions based on its value
  - A value can be a state of a state-machine, and can be easily visualized on a GUI
  - Multiplication of such data points can be quickly done
  - It generates many scripts (C-like interpreter language)
- **Actually it is a set of SMI++ and PVSS and DIM packages**
  - Run control is written by another script language of SMI++
  - PVSS is good at abstract objects and pre-defined PLCs. To control real devices, one has to write C++ code (DIM).
  - According to Clara, NSM looks more like DIM, and to control our own device (e.g. COPPER), we need something like DIM under PVSS. I have no info about it, but probably it's not so trivial to rewrite existing NSM codes to something else

Constructing a system

- **LHCb's control system consists of 150 PVSS systems, ∼50 for DAQ, ∼50 for HLT, ∼50 for others**
  - Belle I needed 15 for DAQ, 2 for HLT and 20 for others (or 400 for DAQ and 150 for HLT including one level lower)
  - Belle II needs 15 for DAQ, 15 for HLT and 20 for others?
- **LHCb control design rules are defined by a single person, and templates are distributed to subdetectors**
  - In Belle I DAQ, most of the systems are unified and used the "same" control program (written by a single person)
  - No particular GUI design rule was there
  - Design rules for control scheme and GUI have to be clearly defined for Belle II, regardless the choice of the framework
  - PVSS probably makes it easier to write a new program based on a given design rules, but PVSS probably makes more work to define the design rules (my impression)

Cost and Maintenance

- **Things were easy at the beginning at CERN when the company was not so strict about license and fee, but it became different after their success at CERN**
- **Cost without negotiation for a large system is rather high according to Christian**
- **No clear idea yet how much we need to rely on company support**
- **I don’t think it’s a good idea to constrain the database choice by PVSS which is coupled with Oracle**
- **So far I’m not comfortable to take it for the run control design as its use is constrained in multiple ways**

PXD run control

- **Components that may be involved: (DHP), DHH, DATCON, ATCA, (GPU), (ROI distributor)**
  - DHP is controlled by DHH, ROI distributor is controlled by HLT?
  - DHH, DATCON and ATCA are controlled by a single PC?
  - Has to connect each other, and to SVD, HLT and EB2
- **Fast control is limited to DHH for now**
  - Both DATCON and ATCA do not have general LVDS I/O port to receive the system clock, so I was thinking that they cannot talk with TTD
  - But slow 254Mbps TRG/ACK signal may be handled by RocketIO. Recovered clock from RocketIO is probably too poor to drive giga-bit RocketIO, but maybe OK to drive 254Mbps ACK. Need to test.
- **Nice to have fast control to DATCON and ATCA for flow control and datalink error monitoring**

My personal conclusion

- **Interaction with LHC control people was quite productive**
  - Now I have a better idea about what is PVSS,
    but I can't say yet that I have a good idea about PVSS
  - Lot of stimulations to improve my idea for the Belle II control system
- **My knowledge on PXD slow control is still quite limited**
  - PVSS may fit better, but we don't have enough information
  - As said, NSM does not provide a "local" control scheme. If the system sequencing can also be made by a short C or C++ program, it also fits the NSM scheme
  - And it is probably easy to control a PVSS based system by a NSM based run control system
  - PVSS does not provide a framework for a fault tolerant system, but it may help to describe such a system better than naive C++
- **Run control design to be started soon, based on NSM**