LIGO

Control State Definition

March 25, 2014

Daniel Sigg, Chris Wipf, Stefan Ballmer

LIGO

Save/Restore

- Relying on save/restore yields inconsistent results
  - Problem 1: “Everyone needs to keep up the snap file”
  - Problem 2: What to restore to?
    Restore to a “good configuration” is a recipe for disaster
  - Problem 3: Restore sometimes skips channels
  - Problem 4: Anyone can make a change without save/restore
  - Problem 5: No good way to tell how actual differs from snap
  - Problem 6: No easy way to make an incremental change
  - Problem 7: It scales badly
  - Problem 8: No easy way to keep subsystems in sync (EX vs. EY)
  - Problem 9: No good way to keep multiple configurations for the same system (e.g. LSC for PRX vs. PRY vs. PRMI, or ASC for PRMI vs. FL)
- Relies on everyone doing the right thing all the time
- Broken work flow!

Filter files

- This is a work flow which functions well
  - Reason 1: All changes are going through a configuration file
  - Reason 2: This is the only way to make a change
  - Reason 3: There is a GUI to make the changes
  - Reason 4: Changes are done incrementally
  - Reason 5: You always know what's running
  - Reason 6: We have a record of old filter files
- The only way to fix a problem is to actually fix it

“New” Approach to Save/Restore

- Divide slow controls channels into 4 groups
  1. Readbacks (ignore for now)
  2. Most of our controls never change
  3. Some change states in a trivial matter
     E.g., boost on when lock bit is set, input matrix for PRX, PRY etc.
  4. Some need to change all the time
- Control State Definition is meant for 2 & 3 (some)
  - Group 4 requires code, i.e., guardian
- CSDef tries to mimic the work flow for filter files

LIGO

Scaling is Important

- Slow channels:
  - ~300,000 slow channels per ifo
  - ~100,000 can be set
  - ~20,000 do change
  - Maybe 10,000 left once you have lookup tables
- Good bookkeeping matters!
- Configuration needs to be duplicated between identical sub-systems
- GUI tool is vitally important to get operators involved
- Commissioning team cannot handle ~100,000 variables

LIGO

Control State Definition

- Configuration file:
  - All slow controls channels must be listed
    - Even, if they are under outside control
    - All unlisted channels are held at zero constant
  - Most channels will be set to a constant value (as opposite to manual)
  - Includes safe and default values
  - Includes lookup tables
    - Most lookup table will have 2 states: “Off” and “Init”
  - Replacement rules for channel names
  - Conditions and Includes for site/location specific configurations
- XML GUI editors are available (schema available)
- Validating parser exists (C++ code)

Control State Definition (2)

- State machine
  - Usual: Init, PreOp, SafeOp and Op modes
  - Will set all values to safe in SafeOp
  - Will set all values to their default when switching to Op
  - Will set values to their configuration when in Op
  - Loads a new configuration file upon request
  - Implemented as a guardian script or as part of the EPICS ioc
- Will initialize all values upon a restart
- Checks constantly while in Op mode
- A value cannot be changed, if it is set constant
- All changes need to go through configuration file
  - Of course, many channels will just be set to manual mode
  - SM watches configuration file and indicates, if it has changed

LIGO

How to support commissioning

- Changing the configuration file is as easy as changing a filter
- Lookup tables have an “Off” state
  - Section a large front-end model into different domains
  - Allows to “talk” in states rather than values, e.g., run/acquisition
  - Allows to gang filter banks
  - Allows for fine grade control
- The state machine can be set into PreOp (no writes)
- Less time wasted to find out “has this changed?”
- Broader user base for keeping up the configuration

LIGO

Why not…

- Hardcode it all in the front-end
  - ECR to change values? Front-end models become very cluttered
- Write an incredibly large guardian script
  - How can you tell what's going on w/o reverse engineering the code?
  - How can you tell that you didn't forget a channel?
- Use hash values in the front end
  - How do you tell what's wrong?
- Use the EPICS access controls
  - This is not a security issue!
- Resurrect the iLIGO Stat system
  - Creates too many secondary channels, CALC records too limited
- Just stay with safe/restore and snap files
  - Proven to be problematic

LIGO

Next on the list

- Alarm and error handling
  - Problem 1: Alarms are global, should be reserved for real problems
  - Problem 2: No clear text messages, why does the IMC not lock?
  - Problem 3: Serious problems go unnoticed
  - Problem 4: Rediscovering the same issues again and again is a major source of wasted time
- Solution with better track record:
  - Condition code pioneered for the squeezer/OAT auto-lockers
  - Hierarchical error structures (error bit, multi-bit code and msg)
  - Each guardian/auto-locker/etc. has a set of conditions which need to be fulfilled to proceed (can be bypassed)
  - Clear text messages of what's wrong
  - Required additional sensors in OAT!