

L1Calo: OKS Cable Schema

Murrough Landon 16 October 2006



- Background
- Implementation
- Suggestions



- There is a complex network of cables connecting parts of the L1Calo trigger
 - Preprocessor to cluster & jet/energy processors
 - Between merger modules in those two processors & to CTP
 - From processor modules to RODs (Glink)
 - RODs to ROS (Slink)
 - ROD busy network
- An early requirement during the development phase (2002) was to be able to simulate arbitrary subsets
 Load playback data anywhere, predict output downstream
- Needed complete description of cabling
 - Preferably easy to edit for frequently changing setups



- Technology: OKS
 - We needed this several years ago at test rigs
 - Chose to extend the OKS configuration DB
 - May want to make another DB the master in future?
- Organisation: separate OKS files
 - Can have file of hardware layout (crates, modules)
 - More than one possible file of cabling layouts
 - Selection of which layout at partition/segments level
 - Though in fact I dont think we ever used this possibility
- Design choices
 - Schema as simple as possible, minimise number of objects
 - Use OKS features to minimise risk of DB editing errors



Actual L1Calo OKS Schema

- One object per cable
 - Or per bundle of cables
- Relates to two modules
- Module connectors are named Cable attributes
 - Strings in base class
 - Enums in subclass for each cable type to help prevent editing errors
- Cable is not HW_Object
 - Was created earlier
 - Separate enable flag





- Custom SW layer on top of DAL
 - Hand written module class with links to DAL objects
 - Also links to calibration & trigger menu data for the module
 - Reverse relationships: modules want to know which cables are connected to them, not vice versa
 - Create multiple connections from one CableBundle
 - Extra sanity checks on top of schema restrictions
 - Check SlinkCable not connected to Busy module etc
 - NB schema only enforces appropriate connector names
 - HW & Simulation classes for each type of module know about the connector names for that module



- Disabling connections
 - Connection is only active if both modules are enabled and the cable object itself is enabled
 - Hardware & simulation mask off disabled inputs
- Experience
 - Very useful to have uniform scheme for all cables
 - But currently breaks down at L1Calo boundaries
 - disabling a ROD doesn't disable the connected InputChannel at the ROS :-(



Suggestions for Wider Use

- Keep it simple!
 - No connector objects
- Cable is HW_Object
- Relationship to Module?
 - Or some other class?
- Use connector numbers?
 - As used in Rack Wizard :-(
 - Would like to keep strings (at least for connector types, some modules have many connectors of different types)
- Generic scheme
 - Extendable to all types of cable, not specific to Busy network and Slinks (otherwise we have to keep our own)





- L1Calo has (slightly specific) schema & SW for cables
- Propose more generic schema for wider use
 - Cables and Modules (or other objects) they connect to are hardware objects (with possible TC references)
 - Objects describing SW can have links to HW objects
- Want to generate OKS DB from other DBs, eg TC
 - But need simple, easily hand editable scheme for test rigs
- May like to have cable subclasses for checking
 - But it gets complex if module relationships are too specific
 - Same cable type can connect different modules, eg Busy input from ROD or another Busy module
- Ensure schema is extendable to subdetector cables
 - And allows customisation with specific "knowledge"