A Large Ion Collider Experiment

Results from the DDS workshop

P. Hristov for CWG13

DDS workshop

Goals

- Learn about the recent DDS development
- Hands on
  - Test the current DDS release candidate
  - Help new users
- Discussions on design:
  - DDS & control system (CS): interactions and requirements
  - DDS & monitoring: interactions and requirements
  - DDS & logging system: interactions and requirements
- Planning: tasks and priorities

DDS & Control System

Meeting outcome

- Functionalities provided by DDS
  - Start processes
  - Stop processes
  - mechanism to send commands (messages) to running processes

- Features provided by Control System
  - Global coordination (start/stop data taking, reconstruction)
  - Send commands to running processes (START, PAUSE, RESUME, CONFIGURE)
  - Skeleton state machine
  - Interactive GUI

- Sample use cases
  - Reset detector Front-End-Electronic (FEE)
  - Include/exclude detector (include/exclude set of FLPs)

DDS & CS

Configuration

- Functionalities provided by DDS
  - Key/value propagation (for keys subscribed at start time)
  - Support for multiple sets of key-value properties defined and used for device configuration. Corresponding devices can subscribe on properties update.
  - Property can be updated either via API, or DDS protocol, or by calling a DDS command line tool. CS can use any of the method to trigger configuration update of devices.

- Features provided by Configuration System
  - Central key-value store
  - API to fetch/change values (on CONFIGURE command)
  - Interactive GUI

DDS & Monitoring

Meeting outcome

- Functionalities provided by DDS
  - Transport layer ? Might be better to use dedicated transport layer.

- Notes
  - Monitoring will use dedicated mon library to allow processes to push values

DDS & Logging

Meeting outcome

- Functionalities provided by DDS
  - Transport layer ? Might be better to use dedicated transport layer.
  - Redirect processes stdout/stderr to logging facility ?
  - DDS messages to logging facility ?

- Notes
  - Processes should avoid cout/printf (do log(…) instead)
  - Logging will use dedicated library to allow processes to tag log messages
  - Should logging messages be treated as monitoring data ?

DDS

Short term plans

- As a proof of concept CS will use DDS protocol to send commands to Tasks (devices) and receive information from Tasks.
- DDS will provide examples on how to create custom channels to use DDS protocol. Using custom channels CS can connect to DDS commander(s) either to request different DDS info or send proxy messages to Tasks.
- DDS will also provide an asynchronies API for User Tasks to give a possibility to subscribe on CS messages. This will be done in a way, like current key-value API is implemented.