

Support for TM's Data Driven Approach

There is a lot of variety contributed by the different types of products that TM will have to interface with in SKA. Hence, it is necessary for TM to optimize the integration cost between TM and product LMC's of SKA. TM takes a data driven approach to solve this and standardizes the way TM integrates with all LMC's. TM implements the idea through the definition of a standard Self-Description (SD) schema that it uses to capture the information about all LMC's in a uniform manner. Each LMC provides their respective information to TM through their respective SD. Following is an initial list of minimum items that will be contained in the self-description:

- a. The logical location details of the Element LMC.
- b. The various generic operating states and the possible transitions that need to be viewed from TM.
- c. List of commands along with their parameters and attributes. This should also contain the possible values, along with constraints for validation.
- d. List of possible responses for the individual commands along with associated parameters and attributes.
- e. List of monitoring points along with associated details such as their rate of updates, interpretations, allowed sampling frequencies, required to be logged and so on.
- f. List of various statuses that the Element can resume, such as Health status levels.
- g. List of events and alarms. This should contain the associated data such as their types, severity levels, recommended handling, frequencies and so on.
- h. Provide description for suppressing of alarms for Elements and their sub-Elements, such as not-fitted, in-maintenance etc.
- i. List of configurations for the maintenance and health checking of the Element LMC such as parameters that TM should monitor.
- j. List of issues that the Element can raise, process to troubleshoot and resolve.
- k. List of events that the Element will need to subscribe to and be provided by TM, e.g. Dish waits for the beam former from the CSP to be ready and send a ready event.
- l. Provide description for any control loops required by the Element, including size of data, source of data, calculations to be performed on data by TM, frequency of updates.

The above information may also evolve going forward. Hence there should be a mechanism in place to implement the SD approach so that all the above information pertaining to an LMC can be captured efficiently and the process supported by appropriate validation and consistency checking. All these information eventually will be consumed by various aspects in TANGO, such as device server, Panic Tool for alarms, Archiver to store the monitoring data and so on. So the process should ideally also allow translating this information into implementation code automatically.

Note:

We saw the need for such a tool in the context of the ITER project. ITER CODAC team implemented the SDD Editor to solve this problem for their context. Inspired by this, we proposed the development of a Domain Specific Language (DSL) for SKA to solve the problem of TM SDD implementation. This has already been prototyped and a demonstration of the same will be provided as a video link shortly.

Primary author: Mr ROY CHAUDHURI, Subhrojyoti (TCS Ltd.)

Co-author: Mrs VAN DEN HEEVER, Lize (SKA South Africa)

Presenter: Mrs VAN DEN HEEVER, Lize (SKA South Africa)