TANGO testing workshop 29/06/2023

We were looking at practical issues experienced when testing TANGO devices in the SKAO. (We solicited more options, but this was what emerged.)

We were looking at https://developer.skao.int/projects/ska-mid-dish-manager/en/latest/_images/DishModeTransitionICD.png as our device state model, with code from https://gitlab.com/ska-telescope/ska-dish-lmc/-/blob/master/tests/utils.py

We ended up with some questions, comments, and advice. The advice is general testing advice. The questions may be trivial to people with more experience.

Questions

1) Is using the –no-db option the same as using a DeviceTestContext in PyTango, or is this different?
2) Can all TANGO devices be modelled as a finite state machine?

Comment

For automated testing of software components only, it's easy to force the state we want. For acceptance tests that involve hardware, managing state is harder - any hints on how to deal with this?

Advice

If interactions with the TangoDB are a problem, you may want to test device-DB interactions separately, before adding more devices to your SUT (System Under Test). (In this specific instance, the DB seems fine, and the problem seems to be in the device.)

For testing state transitions, use events, and wait for the outcome (and set a timeout). If using simulators, we can put sleeps ni there to mimic behaviour of hardware that doesn't respond quickly; ideally, this would be configurable, to allow us to trigger faults/errors from exceeding timeouts.