

Future State of SNCSEQ

Ralph Lange

State Notation Compiler and the Sequencer

- One of the oldest parts of EPICS Base (mentioned before 1989)
 - Generated compiler, lex/yacc-based, for the “SNL” domain specific language
 - Used by almost every EPICS installation
- High-quality code, well documented
 - One of the best-tested modules of Base
 - Ample, complete, detailed, well-written documentation (<https://epics-modules.github.io/sequencer/Manual.html>)
- *Very stable*
 - Last relevant commits more than 5 years ago

The Current Situation

- Ben Franksen recently stepped down as maintainer
 - After ~15 years !!
- SNCSEQ only supports Channel Access
 - Supporting PVAccess would require a lot of effort
- Finite State Machines don't scale to current installations' sizes
 - SNL is fine for close-to-device state machines with 10s-100s channels
 - High-level sequences connecting to 1000s-10000s of channels need other tools

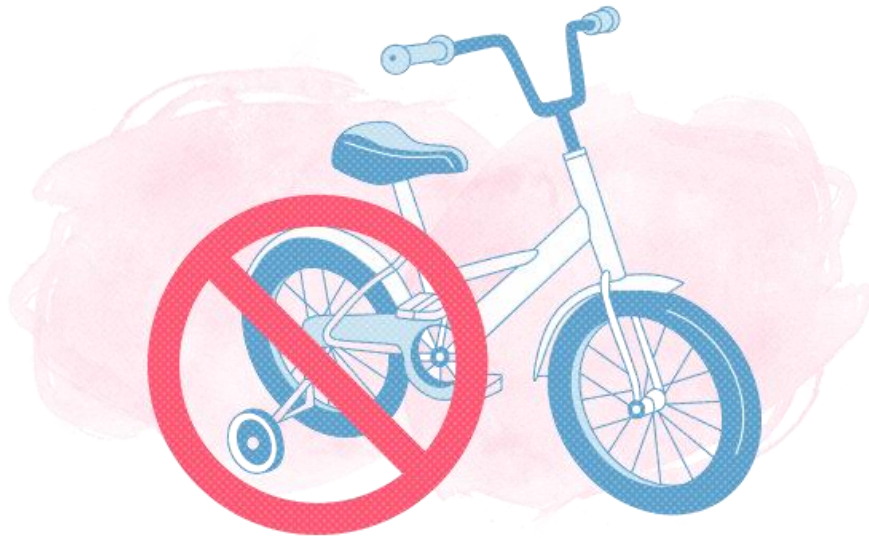
Detour: Complexity Bias

- People tend to choose more complex ideas over simple ones

The situation is very complicated...

The solution is obviously very complex!

- Good example: Options to teach kids how to ride a bike



training wheels



running bike

Idea: Remove Support for Channel Access

- Connect the SEQ runtime directly to the EPICS Process Database
 - The state-machine-side interface could be very similar
 - All connections made at boot time: more deterministic startup
- Clarify the scope for SNCSEQ: on the IOC, close to the device
 - Connections to remote channels possible through local proxy records, supporting everything the EPICS Process Database supports
- Reduce the code base and complexity, improve maintainability
 - SNCSEQ could easier survive the next 30 years...
- Contractable
 - Well-defined project
 - Many existing tests to support the verification

Lightning Talks Get No Discussion

Oooooohhhh...

But obviously, such a step would break many existing applications.

This needs to be discussed.