

## Participant:20 - Offline Answers

### State Machines

*> Since the formalism does not enforce mutual exclusivity of guards, we discuss that it is up to the modeler to handle such cases. Would you agree on this, or do you think that there should be a default behavior if both g1 and g2 evaluate to true?*

I think there should be a default behavior.

*> In the case where none of the guards evaluate to true, should the handling of a default path (e.g., remaining in the same state or transitioning elsewhere) be considered a design decision at the modeling level (i.e., by purposely adding a "!g1 & !g2" guard ), or should it be left as an implementation-specific concern?*

I think a design decision at the modeling level. Otherwise, it might be confusing and error prone.

*> While S is active, it can be re-triggered by event e1 via a self-transition (i.e., S action is executed while e1 keeps occurring). State S exits only when event e2 occurs. Would you agree that this interpretation is correct?*

This is a plausible interpretation, yes.

*> We discussed the possibility of handling errors that span multiple states by introducing composite states that encapsulate all states concerned by that error condition. Would you consider this a suitable modeling approach for representing global or cross-state errors?*

I like this idea. Depending on the complexity that it introduces, I think this could be a suitable modeling approach.

*> Following your remark, we are considering the use of history states (i.e., refining the concept of the mentioned ad hoc states) to support pause-and-resume behavior. Do you agree that introducing history states would be an appropriate modeling solution for this requirement?*

Yes!