1450.4 meeting minutes - 07/10/08

Attendees: Jim O'Reilly, Ajay Khoche

Not present: Doug Sprague, Ernie Wahl, Bruce Parnas, Jose Santiago, Kevin Coggins

Agenda:

- Preamble:
  - Record Meeting (*2) (call not recorded, since we did not have a quorum)
  - IEEE Meeting Preamble (No discussion of proprietary information).
- Call was rescheduled to start ½ hour later than normal. As a result of that change not reaching many of the group in time, only Ajay Khoche and Jim O'Reilly were on the call at 1:00 pm PDT/4:00 pm EDT. Ernie Wahl had called in at the original start time (12:30 PDT/3:30 EDT), and not finding anyone else on the call, departed after about 20 minutes. With both Jim and Ajay on the rescheduled call, we waited for about 20-25 minutes for others to call in. When no one else did, we ended the call.
- Prior to ending the call, however, we discussed two related issues – the SetBin, SetBinStop, and Stop action statements which can appear in various action blocks of TestBase, TestType, FlowType, Test, and Flow, and the proposed STIL .4 TestBase and FlowNode defaults.
- In the current proposal (D22), the actions which can appear in PreActions, PostActions, PassActions, or FailActions blocks of TestBase, TestTypes, Tests, FlowTypes, or Flows, or the actions which can appear in the *exit_port* statement of a FlowNode's **ExitPorts** block both include the SetBin, SetBinStop, and Stop statements.
- Ajay and I discussed the implications of including the Stop statement (either separately, or as part of SetBin – using the SetBinStop statement) as one of the default Fail Actions in TestBase – where it would apply to TestTypes, Tests, FlowTypes, and Flows, unless otherwise overridden by specific Type declarations or object instantiations. We both felt that the Stop statement should NOT be used as the default action for TestBase (and thus for TestTypes or FlowTypes), for the following reasons:
  - In all languages we've reviewed to date, the Stop action (or its equivalent) is a function of the flow sequencing, NOT the Test itself. The Test is simply a set of operations carried out on the DUT, but carries no information about how several Tests should be sequenced, and how to decide what to do if a given Test passes or fails.
  - The test can be executed in isolation from any other Test by a GUI test editor/test execution utility. When doing so, there's no notion of Stop in the context of stopping flow execution. The test simply indicates pass or fail (or some other execution state).
  - Within graphical flow editors, it's easy to represent the Stop action when it's part of the flow. It's less easy, and less intuitive, to figure out a way to represent the Stop action if it's part of the Test itself. Graphical Test Editors simply show the various parameters and execution methods for that test only (and as I mentioned above, allow execution – perhaps repeated execution – of the Test).
  - If we DID allow the Stop statement in the actions of a Test or Flow (as opposed to the exit port actions of a FlowNode), we'd need to reach agreement on the semantics of the Stop statement in the Test, when one might also appear in the FlowNode. My initial thought would be that the complete sequence of events comes to a halt – that is, there would be no return of execution control to the flow node execution engine. In such a case, then any exit port actions of the FlowNode would not be executed. Instead, we'd proceed immediately to end-of-test processing, which would certainly include resolution of binning, summary and/or wafer map updating, etc.
  - I believe that one of the goals we should reach for in STIL.4 is to have our defaults map as closely as possible to current industrial examples of such languages. We can certainly allow the Stop statement to be part of the Test or Flow actions, but I don't think it should be the default action.

- Therefore, I propose that the TestBase defaults do NOT include the Stop statement (either by itself, or as part of the SetBinStop statement. I'd like to discuss the new proposal during this week's meeting.

Next meeting:

- Next meeting 07/17/2008.

For reference STIL .4 information can be found at the IEEE STIL website:
http://grouper.ieee.org/groups/1450/ (select the P1450.4 link from the table) or use the direct link
http://grouper.ieee.org/groups/1450/dot4/index.html