

## **P1450.4 meeting minutes - 02/25/04**

Attendees: Dave Dowding, Tony Taylor, Ernie Wahl, Tom Micek, Jim O'Reilly, Don Organ, Jose Santiago, Jim Mosley

Not present: Yuhai Ma, Doug Sprague, Eric Nguyen

### **Agenda:**

- Update from Tony Taylor on P1450.1 (and other PARs)
- STC/STIL collaboration update (from Jose)
- Review latest flow diagram documents
- Review issues for face-to-face meeting

### **Update on P1450.1:**

PAR for P1450.4 is on NESCOM agenda - their next meeting is in about a month. Expect no problems in getting PAR reapproved.

P1450.1 was two people shy of 75%, and has been extended 10 days. There were some NO votes, so there will be a reballot.

Many issues to resolve - especially around expression syntax.

List of issues is available at (get link from Tony).

P1450.6 is in same state - completed ballot with just over 75%. There were some NO votes, so .6 will reballot as well.

### **Progress on STC/STIL.4 collaboration:**

Update from Jose Santiago: Approval has been given to release (only) the OpenSTAR Test Programming Language (OTPL) documents. This release is pending final review of the documents by the STC working group (making sure, among other things, to remove any references to Advantest) prior to release to the STIL .4 WG. The document should be available to us by the time of our face-to-face meeting on March 16 and March 17.

### **Face-to-face meeting:**

Meeting is finalized for March 16-17, 2004. Location will be at Agilent's campus in Santa Clara, Bldg 54. See Dave Dowding's email to the STIL .4 reflector from Wed 2/25/04 9:37 AM (SUBJECT: stds-1450.4: Working Group Face-to-Face Meetings in the Bay Area 16 and 17 March 2004 Information) for more information.

Definite attendees: Don Organ, Jose Santiago, Ernie Wahl, Dave Dowding, Jim Mosley (or Dan Fan), Tony Taylor

Possible: Jim O'Reilly, Tom Micek

We agreed that it would be helpful to have exposure to Stylus as well as OTPL during the face-to-face meeting. Don has agreed to provide some updated information by the time of our face-to-face meeting. He'll summarize the syntax examples from the Steve Cannon demo of a few weeks ago, and put some documentation around those examples.

### **Discussion of latest flow diagram documents:**

Discussion about the content of Dave's latest flow document (TestProgramFlowWorkingDiagramsB.pdf, dated 2/11/2004) - emailed to WG members on that date (not yet posted on the web site).

- Clarification about the term "Harness" - as seen in Fig. 2 (p5) and Fig. 4 (p6) of the above-mentioned document). As used here, the term "Harness" refers to a data type (or object type) from which the various types of TestModules shown in the figures are derived. In the figures, there are two types of TestModules shown as being referenced by a FlowNode ModuleRef - one which contains a simple VOH test method, and another which contains a subflow. The "Harness" is the common denominator of these two types of TestModules - though we haven't yet defined exactly what that commonality is!

- From last week, there was a discussion (see last week's minutes for details) about whether the module referenced by a FlowNode ModuleRef should be called a TestModule or a FlowModule. After some discussion, we decided to use the term "TestModule".

**Issues to discuss for this week (and in coming weeks):**

- Discussion about what exactly constitutes a "Harness" (**Jim O's comment:** do we want to stick with that term? It seems a bit vague to me. Perhaps we could call the type "TestModule" and the instances TestModuleInstance, or something similar . . .).
- Binning, variables (scalar or multi-dimensional).
- Distributing functions of PatternExec among the new constructs we're defining. Currently, PatternExec specifies the following:

```

PatternExec (PAT_EXEC_NAME) {
    ( Category CATEGORY_NAME ; ) *
    ( Selector SELECTOR_NAME ; ) *
    ( Timing TIMING_NAME ; )
    ( DCLevels (DC_LEVELS_NAME);)
    ( DCSets (DC_SETS_NAME);)
    ( PatternBurst PAT_BURST_NAME ; )
}

```

The specification of each of these will probably be redistributed to either the flow-node level, the TestModule level, or the TestMethod level (or some combination of the above). It might also be possible to allow (for the very simple programs) specification of the above constructs at the test program level.

- Use of and updating of symbolic variables (spec/category/selector variables).
- How to create new (user-defined) types from the basic elements provided by STIL.
- Definition and usage of test program variables, such as
  - Global variables (global in scope to the TestProgram)
  - Other external variables (i.e. TSEM variables)