

1450.4 meeting minutes – 07/27/11

Attendees: Ajay Khoche, Markus Seuring, Jim O'Reilly

Not present: Ernie Wahl, Oleg Erlich, Paul Reuter

Agenda:

- IEEE Meeting Preamble (No discussion of proprietary information).
- Continue discussion of section 7.6, Variables and Expressions. Review recommendations from past two meetings, which had no quorum.

Summary:

- Since the past two meeting had no quorum, but had only recommendations made (which were tentatively agreed to by Ernie and Jim), this week's meeting will review those recommendations with other WG members, towards possible agreement by quorum.
 - Line numbers with respect to Syntax doc dated Jul 20, 2011
- Lines 2109-2184:
 - Per recommendations in 7/13 meeting (included below)

- Line 2080 – 2083 (ref 7/12 draft ballot document):
 - Change Limits operator Contain to return enumerated integer FailMode (see below)

```
1      Enum FailMode {           // Global namespace
2          PASS,
3          FAIL_HILIM,
4          FAIL_LOLIM,
5          FAIL_UNITS,
6          INDETERMINATE,
7      };
8
9      Limits lims = (0s < N < 10ns); // Global namespace
10     failed = lims.Contain(testresult); // Inside TestType
```
 - Allow users to add to FailMode enumerations but not change existing
 - Select a better operator name than Contain ? e.g., Try? CheckLimit? To be discussed; for now, will use Contain.
 - Select a better enumeration name than PASS ? e.g., NONE ? To be discussed; for now, will use PASS

- Agree with having limits check operator, and with having that operator return an enumerated type. However, Ajay feels that proposal is incomplete, is not comfortable with the operator names Contain and NotContain, and pointed out that the Limits row of Table 4 (line 2068, ref 7/20 draft ballot document) is incomplete, containing only the comparison operators `<=` and `<`. He suggested the following:
 - Have the limits comparison specifier (i.e., line 9 in example above) also be able to specify whether or not a result is not only above or below a limit, or inside or outside a limit window, but also whether a result is equal to a particular limit. In addition, he'd like to extend the enumerations to include FAIL (to be applied if result was not equal to a limit, or the limit check otherwise failed, but the user did not care to know HOW the limit check failed – i.e., in a production mode). The details of how FAIL would apply in conjunction with FAIL_HILIM, FAIL_LOLIM, or FAIL_UNITS still need to be worked out, and we'll try to do that in the upcoming meeting this week.
 - Extend the Limits row of table 4 to include the following - `<, >, <=, >=, ==, !=`
 - Change the name of the checking operator from Contain and NotContain (which imply a window) to something more general. Check(...) was suggested. NOTE: the meeting minutes of 7/13 had raised this issue as well – but until we had more people on board at the meeting, we chose to leave it as Contain.
 - All these issues will be discussed at the upcoming meeting this week on 8/3

Actions:

- Jim to rework Verigy example from STIL paper so that the STIL translation is consistent with its original Verigy code.

Reference documents (If logged into your google account, can edit. If not, can only view.)

- <http://spreadsheets.google.com/ccc?key=0AoKiPr1I9LY9dF95dkhSTVVqOU5GbWJyWFNhY0JPX0E&hl=en>
- Namespace resolution examples document:
<http://docs.google.com/Doc?docid=0AYKiPr1I9LY9ZGY4dmNjNTNfMGZkOGJ2bmZy&hl=en>
- Scratchpad spreadsheet: <https://spreadsheets0.google.com/ccc?key=tQ93VDnAZ-C19RfKpPrPDzw&authkey=COzyro8K&hl=en&authkey=COzyro8K#gid=0>
- Scratchpad "Word" doc: https://docs1.google.com/document/d/1zVu2M8nTJsrn0nFbBhiuM8-YRt4ErYqdy_uSa3x3_T4/edit?authkey=CLrgwrsG#

Next meeting: 08/03/11

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>