1450.4 meeting minutes – 05/11/11

Attendees: Ernie Wahl, Paul Reuter, Jim O'Reilly
Not present: Oleg Erlich, Ajay Khoche, Markus Seuring

Agenda:

- IEEE Meeting Preamble (No discussion of proprietary information).
- Continue work on section 7.6 "Variables and Expressions".

Summary:

- Accepted:
  - function "eval" is no longer supported
  - STIL.4 shall not permit partially quoted expressions, mathematical or boolean,
    e.g.:

    0.9*'prd' // Illegal
    '0.9*prd' // Legal

    '0.9*prd' > 10ns // Illegal
    '0.9*prd > 10ns' // Legal
  -
- Discussed, but not resolved:
  - Should the rule that one parameter may notreference an other remain in place ? Considerations we are presently aware of are listed below under PRO, CON, and NOTE. Please inform the group of additional considerations, if any.
    - PRO (desirable effect if it stands):
      - no Test instantiation initialization sequence problems
      - regardless of parameter initialization list order, two tests with the same parameters behave identically.
    - CON (undesirable effects if it continues to stand):
      - Counter-intuitive code behavior:

```
1 Variables {
2       Const Seconds s0 = 5ns;
3 }
4 TestType XXXX {
5       Parameters {
6               In Seconds s0 = 10ns; // Default
7               In Seconds s1 = s0/2; // Default
8       }
9        ...
10 }
```

      - At line 7, the s1 default value is set to 2.5ns, or if line 2 didn't exist, the parser would presumably report an error.

    Unable to implement useful code:

```
1 TestType Histogram {
2       Parameters {
3               In Limits lims;
4               In Compound increment { Units lims.Units; } // Illegal
5                ...
6       }
7       ...
8 }
```

- Line 4 is illegal under current restrictions therefore compile time check for matching units between args lims and increment is not possible. The test intent is to start at the low limit and increment until the test value is no longer within limits.
- NOTE: if we decide that one parameter may reference another, then we need to specify the sequencing mechanism. These are the choices I see:
- sequence is as read at the point of instantiation: same test-type with same parameters may act differently depending on sequence
- sequence is as specified in the TestType: same test-type with same parameters will perform the same but TestType definition must anticipate potential interdepencies or lose some utility.

Actions:

- All WG Members: Please review issues regarding proposed rule change regarding allowing one test/flow type parameter to be defined in terms of another (earlier-defined) parameter. Be prepared to discuss at next meeting.

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-Cl9RFKpPrPDzw&authkey=COzyro8K&hl=en&authkey=COzyro8K#gid=0
- Scratchpad "Word" doc: https://docs1.google.com/document/d/1zVu2M8nTJsrm0nFbBhiuM8-YRt4ErYqdy_uSa3x3_T4/edit?authkey=CLrgwrsG#

Next meeting: 05/18/11

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