1450.4 meeting minutes – 03/25/11

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

Agenda:

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

Summary:

Since the Track keyword proposal was not adopted (due to lack of consistency between STIL.0 and STIL.4 mathemetical expressions and variables), the group wanted to find another way to achieve the benefits of the Track keyword while maintaining the above-mentioned consistency. As described below, the introduction of a mathematical function "eval( )" does achieve that.

From Ernie's notes:

Goal: To define a variable in terms of another variable, such that the defined variable doesn't track and may be reassigned another value. The Track keyword accomplished that but didn't allow for consistent syntax and semantics for STIL.0 and STIL.4 mathematical expressions and variables. The addition of proposed mathematical function "eval" allows for STIL.0/STIL.4 consistency and supports the previously unsupported behavior. The illustration below shows how Integer i achieves the goal:

```
Variables {
          Integer x = 1;
          Integer y = x*2;       // Stores x*2
    Const Integer z = y;         // Stores 1*2 or 2

          Integer i = eval(x*2); // Stores 2
}
// In some actions block:
{
    eval(y);  // Returns 2
    eval(z);  // Returns 2
    eval(i);  // Returns 2
    x = 2;
    eval(y);  // Returns 4, tracks
    eval(z);  // Returns 2, doesn't track but can't be reassigned
    eval(i);  // Returns 2, doesn't track and can   be reassigned
    i = 5;    // Assignment legal for i but wouldn't be for z
}
```

Eval forces inline evaluation (under normal circumstances tracking behaviors can be delayed until an evaluation is required, e.g., statement i < z).
The mathematical expressions on the right-hand side of STIL.0 Spec/Category/variable definitions behave like STIL.4 expressions so eval isn't and doesn't need to be recognized by STIL.0. The STIL.0 Spec/Category/variables behave like non-Const STIL.4 variables which, since STIL.0 doesn't recognize keyword Const, is what their definition syntax looks like.

I've taken this opportunity to add a table of functions to the syntax document including eval, min, and max. Note that min and max are already part of STIL.0 (Table 5, section 6.14, p63 (printed)) and STIL.1 (table 7, section 5.5, p16). That raises the question: do we need to support other functions, e.g., sin, cos, tan, etc. ?

Actions:

- All WG Members: Please review the "eval" proposal and cast your vote (before next week's meeting) on whether or not to include it. If you're unsure or need additional clarification, please request it via email before next week's meeting. Also, if there are other functions that should be included, such as suggested above, make your suggestions to the WG.

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: 04/1/11

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