1450.4 meeting minutes - 12/04/08

Attendees: Jim O'Reilly, Ernie Wahl, Bruce Parnas, Ajay Khoche

Not present: Doug Sprague

Agenda:

- Preamble:
  - Record Meeting (*2)
    - To listen to the meeting recording, do the following:
      - Call the (US) dial-in numbers 1-877-421-0003 (toll free) or 1-770-615-1374 (toll)
      - Enter the passcode code 747464
      - Once dialed in with the proper access code, enter *3 (star 3)
      - Then enter the file number 15587401 for this conference (this number will change each week).
  - Press 1 to listen to the conference
  - IEEE Meeting Preamble (No discussion of proprietary information).
- Logistics: Need to establish new meeting call-in capability. Can't use Doug's IBM conf. number after the end of this year. Would like to have record capability, if possible.
  - Bruce to set up a call-in number with Advantest. Has gotten permission to do so; need to have it set up for calls starting in January. Number will be published when available.
  - Also, starting with the 1/8/2009 meeting, the meetings will last 1 hour, rather than 1 ½ hours. The new meeting times will be 1:00 pm – 2:00 pm Pacific time (4:00 pm – 5:00 pm Eastern time).
- Discuss plans for the remainder of December (WG member availability).
  - Will meet next week (12/11/2008).
  - Will not meet the weeks of 12/18/2008, 12/25/2008, 01/01/2009. Will resume calls (at new call-in number) on 01/08/2009
- Continue review of open issues list. Issues list can be found at http://spreadsheets.google.com/ccc?key=pEI1-gPUmt2ZTw_kcCTgnKw&inv=jim_oreilly@ieee.org&t=93304845348855187 1&guest
  - Attributes for the variables, such as permissions (RO, RW, and RRW proposed), owner (User or System proposed), and description (a text string used to display tooltip-like popups in graphical editing software). These are issues #2 and #5 from the issues list.
    - These are typically needed by test program generation software which allows the user to fill in the blanks via screen entry or GUI software, from which STIL source is generated and can then be converted to a target tester.
    - After reviewing the capabilities of the STIL "Annotation" statement, it was determined that the Annotation statement did not provide sufficient capability. Per Greg Maston, an Annotation statement is associated with a block, and is guaranteed to remain associated with that block; however, the location of such a statement in a block is NOT guaranteed to remain unchanged. So using the Annotation statement as a comment of sorts to apply to a specific statement within the block is not possible.
    - Therefore, we'll extend the attributes of variables to include the above-mentioned items (Permissions, Owner, and Description). There IS precedent for this already in dot1 (the Integer variable type can have InitialValue and Usage attributes, and the SignalVariable type can have InitialValue, Base, and Alignment attributes.
    - We may want to relook at our syntax for variable initialization and for array length declaration – perhaps we should simply adhere to the model set out in dot1, using attributes (identified by specific keywords) to handle these, rather than the more C-like syntax we're now using.

    - Determining the size of an array (issue #11). For an array variable of <name>, the notation <name>.size will return, as an integer, the number of elements in the array. It will NOT specify the amount of storage needed by an array element (for instance, with an array of strings, one can determine how many strings are in the array, but NOT the length of each string).
    - Namespace/scope resolution issues (issue #12). What happens if a local variable (or parameter) has the same name as one at an outer level of scope? Decision: as per C rules, local variable hides the global variable. However, if it's possible to determine the scope of the non-local variable (i.e., by referencing it by its block name), it can be read or written using the complete scope syntax (typically, something of the form <block_name>.<element_name>).
        - Do we want to specify what a compiler or reader should do (i.e., issue a warning or error) if such hiding were to occur? Decision: NO.
        - Scope at any level includes parameters of that level (test or flows), variables at that level, as well as TestProgram global variables. It will NOT include an unnamed global variables block. All other STIL blocks (Timing, levels, etc.) are defined at the top-level, and are global. If there's a Period defined in a spec block, and there's a local variable or parameter named Period, then the local Period would hide the spec block Period.
- Category/Selector/Spec cross references to alter testing based on a measurement result (issue #10). Need to understand the rules in other parts of STIL for how to access (for reading current value, or updating value) of a spec variable. Issue still under study. If sufficient semantic rules don't exist in other parts of STIL, we'll need to create them.
- Next Meeting 12/11/08.

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