1450.4 meeting minutes – 03/18/10

Attendees: Ernie Wahl, Bruce Parnas, Jim O'Reilly, Greg Maston, Markus Seuring, Oleg Erlich

Not present: Ajay Khoche

Agenda:

- IEEE Meeting Preamble (No discussion of proprietary information).
- Minutes from last week are not yet on the web.
- Discussion items.
    - Integration of dot4 variables with dot1 variables
        - How do we merge dot1 variables (local and global) with dot4 variables?
        - Can TestProgram variables be visible within dot1 and dot0 constructs? (PatternExec, PatternBurst, et al)
    - Resolution of above: Three alternatives presented:
        - Dot1 variables (global and local) are completely separate from the dot4 TestProgram global variables namespace, and there's no way to access the dot1 variables from the dot4 namespace.
            - Seen as restrictive – not sure of all the situations in which access to the dot1 variables from the dot4 namespace might be useful, but no one wanted to disallow it.
        - Dot1 global variables are part of the dot4 TestProgram variables namespace.
            - Applies to dot1 global variables only; access to dot1 local variables not permitted. Seemed a bit restrictive; and some were concerned about clashes of variable names and scope (local hiding global; using global without realizing it), leading to unanticipated consequences.
        - Dot1 variables (global and local) are completely separate from the dot4 TestProgram global variables namespace, but can be accessed from the dot4 scope using a block hierarchy specification syntax
            - Somewhat of a philosophical departure from our dot4 guideline that local variables are visible only within the local scope (a la C++ class private data members), but workable.
            - Some ambiguous issues regarding dot1 global variables cleared up by Greg.
                - Uninitialized global variables that are set to some value in one PatternExec do NOT carry over that value to the next PatternExec. Variable becomes uninitialized at the start of the next PatternExec.
                - Global variables are, in effect, local in scope to a PatternExec. It's as if one specified a domain-named Variables block within the PatternExec scope.
            - Solution is workable, but requires that PatternExec blocks be named in order to allow access to dot1 global variables (which are, in effect, local in scope to PatternExec and below). Not a real problem, as dot4 already requires that PatternExec blocks be named.
        - After much discussion, we chose the third alternative. We'll develop a block-hierarchy access syntax such as

          PAT_EXEC_NAME.PAT_BURST_NAME.PAT_LIST_NAME.**Variables**["VAR_NAME"]

          to allow access (read and write) of dot1 variables (global and local) from the dot4 namespace. Rules will need to be written to specify what happens if a variable is written from the dot4 namespace, when it either has an initial value in the declaration, or it doesn't. In the case where it does have an initial value, writing to it from the dot4 namespace has no effect, since at the beginning of the

PatternExec (or PatternBurst, or PatternList) will overwrite the dot4-provided value.
        - We'll also develop syntax to extend the PatternExec block to allow specification of instantiating domain-named Variables blocks in the PatternExec block. Using this form instead of the unnamed (global) variables block would provide a clearer indication of intent. The reference of domain-named variables blocks in PatternExec would behave exactly as unnamed variables blocks do now, and the two could coexist (barring any variable name clashes).
- Adaptive test – do we want to explore dynamic reordering of PatLists or ParallelPatLists?
    - Add to Issues/resolutions list.
- Open issues - are there other open issues that should be considered? A review of the open issues list can guide us here.
    - Issues list: http://spreadsheets.google.com/ccc?key=0AoKiPr1I9LY9dF95dkhSTVVqOU5GbWJyWFNhY0JPX0E&hl=en
    - Namespace resolution examples document: http://docs.google.com/Doc?docid=0AYKiPr1I9LY9ZGY4dmNjNTNfMGZkOGJ2bmZy&hl=en
    - If logged into your google account, can edit. If not, can only view.
- Next Meeting 03/25/10.

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