P1450.4 syntax subgroup meeting minutes – 10/10/05

Attendees: Jim O'Reilly, Dave Dowding, Doug Sprague, Greg Maston, Tony Taylor

Not present:

Agenda:

- Discussion about SB Thum's role
  - Agreed that SB's assistance would be most useful immediately in refinement of the conceptual model and syntax use cases – the latter which could be discussed in the Monday syntax subgroup meetings, at which SB is welcome to attend.
  - Dave to responsd to Chris's request/offer via email.
- TestMethods – (i.e., "dot5")
  - Comments from WG members (Chris Nelson, et al) that there's no real benefit to standardizing the TestMethods (except for the interface mechanism to .4)
  - The STIL User's Group seems a more appropriate venue for managing the test method library definition (and descriptions – to an appropriate level of detail – about what eac test method does).
  - This issue should be discussed with the full WG at the Wednesday meeting.
- Discussion about suggestions from Wednesday meeting.
  - At Wednesday's meeting, in a discussion about the behavior of Variables in the TestProgram, TestFlow, and TestNode blocks (and the scope rules which pertain to the variables in those blocks at each level), it was pointed out that global variables are also needed.
    - There currently exists a way to allow global variables – if an unnamed Variables block is defined, those variables are created automatically – and are global in scope.
    - Further, we agreed that the "**Variables** (VAR_DOMAIN);" statement in the TestProgram block will create a variables block that is global in scope to the entire TestProgram block. Conversely, the "**Variables** (VAR_DOMAIN);" in either the TestFlow or the TestNode blocks will create variables local in scope to each instance of that block.
- Discuss inheritance
  - Agreed that the inheritance mechanism, and the semantics summarized by Tony in the D15 (Oct. 3, 2005) draft be accepted as written. The issue will be reviewed before the full WG.
- Discuss execute statement.
  - The execute statement has two different forms:
    - One in which some or all of the values of parameters for the object being executed can be specified at execution time.
    - One in which the execution uses parameter values as specified in either the type definition, or (if an instance) at the time the instance was created.
  - The question was: do we want to allow the 2nd form to be used with previously created instances, or only with unnamed instances created on-the-fly by the TestExec statement?
    - The former would mean that an an instance's parameter values could be modified at execution time – which kind of defeats the purpose of having an instance.
    - The latter means that an instance's parameter values are fixed at the time the instance is created – whether that is done before use via the TestInstances statement, or on-the-fly at point of use via the TestExec statement.
    - In summary, we decided to **disallow** the second form (which specifies parameter values at point of call) for (previously-created) TestInstances, and only allow it for of a TestNode, TestFlow, or TestMethod)
- Discuss the contents of pre and post actions for TestNodes and TestFlows, as well as the contents of exit port statements

- Reviewed the syntax put forth in Jim's document dated Sept. 27, 2005 ("Proposal for changes to IEEE P1450.4/D15– Sept. 27, 2005"
- Agreement (I think) on the need for slightly different forms of the exit statement for TestNodes and FlowNodes.
- General agreement on the content of the constructs – but the specific syntax still needs work.
    - In particular, the use of the keyword OnCondition <bool_expr> to represent the arbiter seems a bit clumsy and verbose – need to specify a separate condition for each port.
    - Also, the use of Exit and Stop as possible actions in the pre-actions and post-actions statement seems misplaced – shouldn't these be in the TestNode or FlowNode exit-port statements?
    - Finally, the issue of how to specify the next flow node in the chain (for FlowNode exit port statements) was discussed – with several good ideas, but none discussed enough to reach consensus.
    - Jim and Tony will work offline to come up with proposed syntax for each of these areas, and we'll discuss it next week.

- Finally, some additional comments from last Wednesday's meeting were discussed.
    - In particular, the use of the word TestNode for the construct to which it's attached. It was felt that the similarity of that term to FlowNode was potentially confusing – some of the suggestions offered last Wednesday were TestType, TestObject, and TestBlock. More discussion is required – but as we want to quickly bring the conceptual model terminology and the syntax document terminology into sync, we'll have to at that point settled on a name for the construct.

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