P1450.4 meeting minutes - 01/10/07

Attendees: Doug Sprague, Jim O'Reilly, Greg Maston, Bruce Parnas, Jose Santiago, Brian Johnson, Ernie Wahl

Not present: Ajay Koche, Tony Taylor, Carol Dowding, Daniel Fan, Yuhai Ma, Bob Roberts, Oscar Rodrigues, Jim Mosley, SB Thum

Agenda/Summary:

- P**reamble:**
  - 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 82399501 for this conference (this number will change each week).
      - Press 1 to listen to the conference.
  - IEEE Meeting Preamble (No discussion of proprietary information)

- **TestBase, TestMethods, and Inheritance Discussion**
  - TestBase will work for Methods and Flows as long as .5 does not define a TestFlow TestMethod.
  - Discussion about current proposal for providing a common set of elements for all TestMethods. At the previous meeting, it was requested that we develop an approach which would provide defaults automatically, even if inheritance is not used. The reason for the request is to avoid biasing the language of the standard toward requiring the use of an OO approach (that is, maintain a separation of the language of the standard from the implementation or usage of that language). Therefore, the current proposal is as follows:
    - TestBase will be defined as a separate STIL block. The materials distributed prior to the meeting showed the allowable syntax for TestBase. The STIL-BNF syntax is very similar to that for TestMethod, but excludes (disallows) the PatternExec and TestExec statements.
    - The standard will take that general syntax, and prescribe a minimum set of elements (parameters, variables, pre- and post-actions, and arbiter). The minimum set of elements has not yet been defined or agreed on, but various proposals have been made, and will be reexamined.
    - This minimum set of elements will apply to ANY TestMethod or TestFlow that does NOT explicitly inherit from another like type (i.e., TestFlows from other TestFlows, TestMethods from other TestMethods). Stated another way, TestBase provides a set of defaults (which can be considered as automatic inheritance, if you wish to think of it in those terms) that applies to TestMethods or TestFlows which don't derive from anything else.
    - The current proposal allows TestBase to be modified.
      - Modifications to parameters or variables MUST include the minimum default set, in addition to those needed by the user. If this condition is not met, the end result is NOT compliant with STIL.4.
      - Modifications to pre-actions, post-actions, and arbiter will REPLACE the default set. If there are actions in the default set that the user needs, these must be explicitly included in the user-specified set.
    - Question: Is it necessary to allow the contents of TestBase to be modified by the user (i.e., the ATE SW developer or perhaps a library maintainer at an end-user site)? Arguments were made both ways.

            - If STIL.4 is to serve as both a native runtime language and as a translation medium to some other source language, it might be necessary to allow modifying TestBase, to meet the constraints of a particular ATE SW, or a particular customer. In this case, it's required that TestBase be a separate syntax block, with minimum required elements specified.
            - On the other hand, if we choose NOT to allow modification of TestBase, it may not be necessary to have TestBase as a separate block type with its own syntax. Instead, the standard will simply enumerate the default elements of TestMethod and TestFlow. These standard elements apply even if not stated explicitly.
            - The issue needs further study. We'll study the situations for which it is claimed that TestBase needs to be modified, and determine if this is indeed the case.
        - There was some confusion as to whether or not the current proposal met the requirement of providing a base set of automatic defaults (which are ALWAYS guaranteed to be present) if inheritance were not used. This confusion, I believe, arose because we simply showed the STIL-BNF syntax for TestBase (which defines the allowable syntax), without showing the specifics of what base elements the standard will require of TestMethods and TestFlows. I believe that if we show these specifics (using the syntax proposed for TestBase), the intent will be clearer. Those specifics will be distributed prior to the next meeting, and discussed at the next meeting.
    - Treating TestFlow as a TestMethod (i.e., TestFlow is the one integral TestMethod defined in P1450.4)
        - There is a need to be able treat TestMethods and TestFlows interchangeably. The reason for this is to be able to create user-defined types which build on combinations of previously-defined types. These user-defined types will typically be implemented by creating small flows; these flows must be callable as TestMethods, just as integral TestMethods are.
            - An example of this is a VOH test which tests functionally vs. parametrically (where the functional test is a single test, while the parametric test is a sequence of tests).
        - The current proposals (syntax and conceptual model) don't treat TestFlow as a type of TestMethod, but instead define TestFlow as a separate block with its own syntax. This syntax is very similar, but not identical to, the syntax for TestMethod. The current proposals allow TestFlows and Test Methods to be used interchangeably in the TestInstances or TestExec statements.
        - There was some concern that the current proposal for TestBase (see above) may not work in this situation (i.e., can you guarantee that the defaults defined by TestBase will also apply to user-defined types?). However, it was felt (and tentatively agreed upon) that the current TestBase proposal will work for TestMethods and TestFlows AS LONG AS P1450.5 does NOT define a TestFlow TestMethod. Since we're defining the TestFlow (whether or not you consider it a TestMethod or not is irrelevant, since a TestFlow and TestMethod can be used interchangeably) within the context of .4, I believe that the stated concern will not be an issue.

Next meeting:

- Next Meeting 01/17/2007.

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