# P1450.4 meeting minutes - 09/15/04

Attendees: Dave Dowding, Jim O'Reilly, Tony Taylor, Jose Santiago, Ernie Wahl, Oscar Rodriguez, Don Organ

Not present: Tom Micek, Yuhai Ma, Jim Mosley, Eric Nguyen, Dan Fan, Jim Mosley

**Agenda:**

- ITC items.
- Review drafts proposals from syntax subgroup. Discuss differences and establish a plan for evaluating them.
- STC update from Jose

**ITC items**

- Dave talking with Rojit Kapur about setting up a WG face-to-face meeting at ITC (need to get meeting room). Monday, 10/25 appears to be a likely good date for such a meeting. Depending on facilities, we may also have a conference call. Stay tuned . . .
- TTSC report. Need to determine dates for the following milestones:
    - Agreement on syntax.
    - Semantics (definitions and descriptions behind syntax).
    - Doc complete (examples and tutorials).
    - Ready for ballot

**Syntax subgroup status:**

- Discussed the two alternative syntax drafts (Tony's most recent D11 document – see Tony's email to group dated Wed 9/15/2004 9:52 AM), and Jim's D13 document).
    - Described the differences between these two proposals (see summary below)
    - Established a plan for evaluating both. We'll take use cases and other examples, and render them into each of the alternative syntaxes (syntaces?). From there, we'll see which approach (or which elements of each) works best.
    - Need to establish defaults for TestNode, TestObject (D13 only), and TestMethod.
- Summary of differences between D11 and D13
    - D11 does NOT contain a TestObject block, but instead has only TestNode and TestMethod blocks (along with TestFlow and TestProgram). Instead, the TestMethod syntax has two alternatives – one which contains only a test method name (the type definition), and another which includes both a test method name (the type definition) and a test method instance name (the name of the instantiated object).

      **TestMethod** TEST_METHOD_NAME (TEST_METHOD_INSTANCE_NAME) **{**
      ( **Inherit** TEST_METHOD_NAME**;** )
      ( <**In** | **Out** > (**Const**) (**Once**) *data-type* < NAME | NAME **[**INDEX**]** > (**=** *expr* ) **;**)* *// test method params*
      **}** ) *// end TestObject*

      If both the method name and method instance name are provided, it's intended that an object of that type will be created, from which the actual test method type can be invoked.

      If only the method name is provided, then no object is created, but the test method type will simply be invoked.

      Note that Tony's latest draft of D11 also contains an optional instance name for the testnode. This is redundant, and is not required. In discussing this with Tony, we agreed that for the test node, the keyword TestNode itself provided the type definition, while the test node name provided the name for an object of type TestNode. Note also that a TestNode block can be created inside a TestFlow, or outside. If created outside a

testflow, a TestNode name is required. If created inside a testflow, the testnode name is optional – it would be needed only if some other testnode needed to refer to it by name (other than Next) as the next testnode in the flow. Note also that any testnodes created inside a testflow are valid ONLY within the scope of that testflow – meaning that no other testflow can make use of that testnode.

Note that this formulation (apparently) does not allow for pre- and post-actions within the TestMethod object (as is allowed in the formulation with TestObject in D13). This issue may need to be addressed, or we could just use the TestNode pre- and post-actions.

Alternatively, it is possible to emulate a TestObject using a TestNode which excludes things like Title, Override, and Position – and have pass-specific and/or fail-specific post-actions, where both the pass and fail path reconverge by both specifying the same next node.

- D13 contains a TestObject block, as well as TestNode, TestMethod, TestFlow, and TestProgram blocks. The TestObject here is an object of some TestMethod type; the TestObject contains many of the elements of a TestNode (pre- and post-actions, an arbiter (of sorts – based solely on whether the test method call passed or failed), but allows only a single exit path. It's a heavier-weight syntax, with more elements than the D11 formulation – which is better remains to be seen.

- Rendering our examples using both syntaxes.
  - Tony - PatternExec example and Inovys example.
  - Jose – can do one of his use cases
  - Jim – can do at least one of the use case 1 and use case 4.

**STC status:**

- Jose – no recent activity; planning a meeting tomorrow (9/16). Focus of discussions recently have been about multi-site – these discussions are expected to continue tomorrow.