

## P1450.4 meeting minutes - 09/28/05

Attendees: Dave Dowding, Jose Santiago, Chris Nelson, SB Thum, Jim O'Reilly, Bob Roberts, Ernie Wahl

Not present: Tom Micek, Yuhai Ma, Eric Nguyen, Steve Lill, Tony Taylor, Daniel Fan, Doug Sprague, Jim Mosley, Oscar Rodriguez

### Summary:

- Face-to-face meetings for various STIL WGs (especially for .4) will be set up at ITC. Get info
  - 11/7 1-5 pm Agilent field office. Dave to send out more details (maps, etc.) as they become available.
  - AMS date? All meeting info will be sent to reflector. .3 – same time as .4 – different location (Freescale facility). Can't schedule these to be non-conflicting.
  - Would be great if we could schedule these to be non-overlapping (morning for .3, afternoon for .4?)
  - Chris - .3 on hiatus? Is the .3 a startup meeting?
  - Dave – will try to reschedule these meetings, and have them at the same location.
  - .3 is NOT on hiatus – it's currently reactivated, with the conclusion of .1 effort.
- Walk through latest version of conceptual model document with full WG. Purpose – to explain the conceptual, gain agreement (or dissent) from WG members on various issues, and align terminology between conceptual model and syntax document.
  - For the time being, as terminology changes, can we include a mapping of new terms and old terms?
  - **For the purpose of these minutes, the terms used will be those from the conceptual model.**
- Comments from WG members.
  - Has Fig. 1 – overall diagram – been deleted? If so, why?
    - Been removed only temporarily – will be updated and reinserted.
  - SB – where does flow decision making happen? FlowNode, or TestMethod (TestNode)? For reference, see Fig. 4.
    - All flow decision making happens in FlowNode. Fig. 4 is out of date and will be updated or removed.
  - SB – can TestFlow be an object?
    - Yes – discussion about the notions of types (TestFlow and TestMethod) vs instances.
  - Chris – how much has changed since 8/29 (the last copy he reviewed and marked up)
    - Why are we revising the document and not changing the version number?
      - Each major draft is to deal with a certain set of issues – once resolved, then we move onto the next set of issues and bump the draft number.
      - As the document is revised while each set of issues is being worked on, the draft number will NOT change, but the document DATE will change.
  - Entry points.
    - Chris Nelson – can we have unbounded user-defined entry points?
    - Ernie – user-defined can cause portability problems – would like to understand the need for user-defined entry points. User-defined entities may cause portability problems.
    - Chris – An example might be 14 variations of user-defined characterization flow – since (clearly?) there's something outside this system controlling the system, there must be agreement between the STIL program (which defines the entry points) and the external controller (which must know about these entry points and HOW to call them).

- Chris - Re: portability – that issue DOES exist today in .0, .1, .2, etc. Can always write legal STIL which can't run on a given tester – which gets you into the .3 (Tester Rules Check) space.
    - Jim – Keep in mind that you can have multiple test programs – each with its own set of EntryPoints. This COULD meet the need that Chris described above.
- Some discussion about integral vs user-defined TestMethods, which included a brief digression into P1450.4 vs. P1450.5, and the intent of .5 (the only TestMethod defined by .4 will be TestFlow).
- Comment from Chris - Dot 5 doesn't make sense from Intel's perspective. Dot 4 should define how to interface to test methods, but doesn't make sense to define the details of a TestMethod without using a full-featured programming language.
  - To deal with the portability issue, wouldn't we need to at least define the TYPES of test methods and their input/output parameters?
  - Perhaps dealing with this via the STIL users' group would be a better approach.
- Continue discussion about integral vs. user-defined types.
  - SB - What are major differences between integral and user-defined?
  - Ernie:
    - Integral types are expected to be provided by ATE vendor.
    - User-defined types will be defined by the user by combining integral and other user-defined methods (generally by using the TestMethod Flow).
  - Chris - is a user-defined type any different than a hierarchical flow? Why call it a new test method?
  - Ernie – the reason for calling it a new test method is so that you can define the type once, and use it over and over again. Integral flow would not have arguments, but user can derive from that flow a user-defined flow which DOES take arguments
  - Chris – Would that require parameterized flows? Ernie – yes.
    - EDITOR'S NOTE: And of course, the ability to create parameterized flows DOES exist!
  - Ernie – describing the difference between an instance of a flow and a type definition of a flow(Type flow vs instance flow).
    - Instances of flow type shouldn't look different from any other test (i.e., closed box).
    - An integral type does not permit any viewing of its internals, whereas a user-defined type can (under certain circumstances, such as an editor for a block of that type being run in engineering mode instead of production mode) permit viewing of its internals. And perhaps there would/should be a flag identifying a type as either an integral type or a user-defined type.
  - SB: User-defined vs. integral test types– this topic is confusing. Need clearer explanation.
  - Discussion about the components of the base type BaseTest
- Discussion about section 2.4.1. SB was asking about the reason for and the behavior of the various port actions. Somewhat longish discussion about how these are intended to work.

#### **Actions:**

- Jim to summarize comments in minutes, and take them back to the syntax subgroup for further discussion/action.
- Dave – we'll pick up on this discussion next week.
- Chris' request – Would like to have some graphics put together to illustrate the differences between integral and user-defined types – a real foundation concept which better demonstration. Highlight the distinctions and the value added by the user-defined type.

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>