# 1450.4 meeting minutes – 03/04/10

**Attendees**: Ernie Wahl, Markus Seuring, Bruce Parnas, Jim O'Reilly

**Not present**: Oleg Erlich, Ajay Khoche

**Agenda:**

- IEEE Meeting Preamble (No discussion of proprietary information).
- Minutes from last week are on the web.
- Questions:
  - Do we want to define in STIL.4 code the behavior of tests, or only specify the interface to those tests?
    - STIL.4 will specify the interface to Tests, and describe the behavior of Tests, but will not provide detailed syntax constructs that may be needed to IMPLEMENT the test. That will be left to either dot5, future dot4 work, or perhaps always remain in the realm of individual implementations.
  - Should .Min, .Typ, and .Max fields of a spec variable be modifiable from within STIL.4 code (as we're allowing for .Meas)?
    - Example: How do we set up a test in which the position of an output (for instance, an output clock edge) is measured, and other timing is set up relative to that measurement?
      - If we can do this without needing to have .Min, .Typ, .Max writable we will. If not, will revisit above.
    - End result: Since we're not going to try to provide syntax for actually IMPLEMENTING tests, but rather just provide the interface to a test, and describe in enough detail to be unambiguous what a specific test type will do, there's no need to consider this further.
  - Can you define spec block selector items (i.e., .Min, .Typ, .Max) in terms of other selector fields items?
    - No. For now, consensus on leaving Min, Typ, Max as constant; .Meas is non-constant and can have a value assigned OUTSIDE spec block definition.
  - Do we want to allow for initialization arithmetic involving parameters and /or variables (when a Test or Flow is instantiated)?
    - Yes.
  - When instantiating a Test or Flow, do we want to allow a later parameter to refer to (or use) the value of a previous one to set the value during instantiation, and what value it sees if it can or can't.
    - No. See rules below:
      - When instantiating a TestType or FlowType, values for Parameters (default and instantiation-provided) will be initialized before any local Variables. This includes any Parameters defined in TestBase.
      - When instantiating a TestType or FlowType, Type default values will be assigned before any values provided at instantiation. There is no requirement regarding the order in which Parameters be initialized. Variables will be initialized in the order in which they're specified in the Variables block of the TestType or FlowType definition.
      - When instantiating a TestType or FlowType, values for Parameters (default and instantiation-provided) will be initialized before any local Variables. This includes any Parameters defined in TestBase.
      - Within a Parameters block, there can be no referencing of other items from within that block. A Variables block, however, can refer to items in the Parameters block – and further, a later-defined Variable can be initialized from a previously-defined one.
  - At instantiation of Test or Flow (from TestType or FlowType) what is the order of initialization of parameters? Per C++ rules, initialized in order defined in type (class) definition. Do we want to use that rule?
    - No. See comments above.

- 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 03/11/10.

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