1450.4 meeting minutes - 01/24/08

Attendees:Doug Sprague, Bruce Parnas, Ernie Wahl, Jim O'Reilly, Ajay Koche

Not present: Jose Santiago

Agenda:

- Preamble:
  - 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 13624501 for this conference (this number will change each week).
      - Press 1 to listen to the conference.
  - IEEE Meeting Preamble (No discussion of proprietary information).
- Website maintenance. Tony backing out of STIL activities (including website maintenance). Need to talk with Greg Maston regarding website maintenance.
  - Some discussion with Greg, need to follow up and finalize. For the time being, will continue the model that Greg and Tony had been using - sharing a login, and having the dot4 website files be located in a subdirectory of the dot0 website. If we set up a separate login and account, we may need to create a separate structure for the website, and just link to it from the P1450 home page.
  - Jim to follow up with Greg by 1/31/08.
- VTS face to face meeting (in San Diego) (all)
  - Doug: maybe; Ernie: probably not; Jim: likely; Bruce: probably not; Ajay: yes
- Dot0 re-ballot plans (Doug)
  - According to Greg, focus on dot0 reballot will be on clarifications of usage, not on adding additional capability. Any additional capability should be added through the extension(s) that address that capability.
- Mechanism for iteration through dot0 specs/categories (Jim)
  - Discussion about how dot0 handles spec block and categories/selectors.
    - Names of spec blocks are NOT referenced anywhere else by dot0 (or any other approved dotN extensions). Spec names and category names are essentially global variables.
    - Any mechanism we come up with in dot4 to iterate over the categories in a spec block MUST work with this constraint.
  - Discussion of how to iterate over categories in a spec block.
    - Issue(s): Usage and semantics of spec blocks within dot0
      - Names of spec blocks are NOT referenced anywhere else by dot0 (or any other approved dotN extensions). Spec names and category names are essentially global variables.
      - Any mechanism we come up with in dot4 to iterate over the categories in a spec block MUST work with this constraint.
    - How to iterate over categories in a spec block (and connect each category to an axis in a bindef block).
      - Which should be the controlling construct?
        - Iterate over categories in the spec block, and connect each with a bin axis, or
        - Iterate over axes in a bindef, and connect each with a spec category).
      - The first seems more logical (that is, the specs/categories are the elements which control the test conditions applied, while the bindef

axis is simply an element which can be associated with a particular set of conditions).

- Should the iteration access be indexed based (i.e., based on the order that the categories appear in a spec block)?
  - Probably should not be based on an index order, but rather the name of the category.
    - There is no dot0 requirement that writers output a list within a block (for instance, categories within a spec block) in the same order that it was initially read in (though in general, it's quite likely that writers will do so.
    - Also, thinking about manual edits of categories within a spec block, or axes within a bindef block. If a user manually rearranges these, or adds a category without a corresponding bindef axis, we'd like some automatic (syntactical or semantic) check that would flag the user.
  - We don't want order specific issues with named blocks. Therefore, (per a discussion with Greg Maston), a name-based approach (in which each category within a spec block might somehow specify which bin axis is associated with that category) is probably better.
- In order for the above iteration to work, we need some way of determining automatically the number of categories in a spec block, and the number of axes in a bindef block. A syntactical construct added to the spec block syntax seems feasible (especially since the name of the spec block is used for no other purpose, and is therefore unlikely to be in conflict with existing usage).
  - Example: SPEC_BLOCK_NAME.numCategories to return the number of categories
  - SPEC_BLOCK_NAME.category[idx] to return the name of a particular category. This category name value can then be applied to the "**Category** CATEGORY_NAME;" statement in the PatternExec block (or other dot-4 defined blocks which include the "**Category** CATEGORY_NAME;" statement
- We may need to consider augmenting PatternExec block to specify spec blocks to be used, along with Category and Selector. Look at existing industrial examples (Envision, ASAP, XTOS, SmarTest, OTPL, and Stylus) to determine how each of these languages handles that issue, compared to STIL.
  - Need to consider whether such a change might break existing code, or otherwise be in conflict. Perhaps such a change could be done only in context of dot4, so that if
    - Ernie's desire is to have some linkage between number of axes in bindef block and number of categories in a spec block.
    - Jim to more fully develop and document the above ideas as a proposal. To be ready for next meeting.

- Ernie – with the usage (or non-usage, in this case) of spec block names as an example, we need to add to our to-do list an item which addresses the issue of dot4 Integration with other P1450.X specs
  - One issues is spec/category access in dot0
- Continued discussion of Ernie's binning document (All)
  - Discussion of counters – how to multiple sets (levels) of counters get indexed, accessed, and reset?
  - Referring to Ernie's binning/counter proposal document**,** starting at 1.6.1, table 6
    - Inc line 6, 1.66GHz counter incremented

            - Causes line 3, ClockSpeed to be incremented
            - Causes line 2, Pass Counter to be incremented
        - Visualize each entry in table 5 has a copy of table 6 within it
        - Should allow query (read) access to table entries through some dot notation, but DON'T allow write access. Updates of counters are handled by the system through the normal binning process.
    - Hard Bins 1.6.2
        - Similiar to softbins
    - Back to table 6
        - Ernie to note in currrent document that we should revisit whether to allow multiple levels of Axes (not multiple, but multiple levels).
            - Issues: Is this really needed, and what burden does it put on the spec
            - Bruce to investigate if multiple levels of Axes in OTPL is ever used by customers
    - 1.6.3. Counter Based Events
- Ernie to send updated document for us to review next time

Action Items:
AI01: Doug to add new high level todolist item "Integration with Other 1450 specs"
AI02: Jim to send proposal for handling syncronization between dot0 spec/cat and dot4 binning
AI03: Bruce to investigate usage of multiple axes for OTPL with Advantest customers
AI04: Ernie to send latest update of binning document before next meeting.

Next meeting:
- Next meeting 01/31/2008.

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

Updated todolist:
---------- Latest High Level Dot4 Todolist ----------
- Binning (Ernie)
- Runtime Variables (Doug)
- PatternExec constructs (Jim)
- Bypass (in PreActions) syntax and semantics (Jim)
- TestBase (Jim)
- Input/Output Interface? (Ernie??)
  - Writing/Reading to/from screen, files
  - Interfacing to Shmoo Plotting
- MultiSite (??)
- Standard Predefined Test Methods (Ernie)
- Datalogging Interface? (Ajay)
- Process (Jose, Doug, Jim)
- Framemaker Documentation (Jose)
- Integration with Other 1450 Specs (??)