

1450.4 meeting minutes – 06/29/11

Attendees: Ernie Wahl, Markus Seuring, , Jim O'Reilly

Not present: Oleg Erlich, Ajay Khoche, Paul Reuter

NOTE: No meeting was held on June 22, 2011.

Agenda:

- IEEE Meeting Preamble (No discussion of proprietary information).
- Discuss semantics of FunctionDefs/FuncExec constructs.
- Continue discussion of section 7.6, Variables and Expressions.

Summary:

- Previous unsettled business: with regard to default spec variable values to be used in the absence of a Selector, see meeting notes 6/8/2011.
- Jim O'Reilly will continue work offline on FunctionDefs/FuncExec.
- With respect to the Syntax Document dated Jun 7, 2011:
 - Line 2263: Consensus: operator Name returns the variable/parameter instance identifier. For InOut parameters (references), Name returns the dot separated hierarchical name of the original data structure, not the local parameter name.
 - Line 1792: Consensus: Note that, for simplicity's sake, a spec variable shall always be treated as mutable since its actual value is Selector dependent.
 - Line 1863: Consensus: There two kinds of constraints: intrinsic and user-settable. Intrinsic constraints are implied by the variable or parameter type, e.g., type Seconds can only be initialized to or assigned a value of type seconds whereas type Compound may be initialized to any numeric value. User settable constraints may occur in multiple locations. Type-modifier keyword Const, which occurs before the type-name, constrains a variable or parameter to be immutable. Other constraints, located between braces along with attributes, restrict legal assignments to a subset of what might otherwise be legal.
 - Line 1869: Consensus: Attributes are specified by one or more semicolon separated keywords. An attribute, if not explicitly specified, takes on its default value. Some attributes allow, e.g., a STIL.4 GUI, to impose constraints. When editing STIL.4 code via a text editor, these attributes merely serve as hints.
 - Line 2038: Under consideration: Parameter <TestType>: used to pass a reference to a STIL.4 test structure, i.e., an instantiation of <TestType> or any type derived from <TestType>. If TestBase is specified as the <TestType>, then this is no constraint at all since alltest-types are derived from TestBase. If StdFunctional is specified as the <TestType>, then only an instantiation of StdFunctional or a type derived from it may be passed in. An application would be a generic histogram generating test-type that loops over the user's chosen test with each iteration.

```
// Histogram code is pared down to cut down on obfuscation, i.e., limits and
// iteration code is omitted.
```

```
TestType Histogram {
    Inherit TestBase;
    Parameters {
        In StdFunctional test; // <TestType> = StdFunctional
    }
    TestExec test;
}
```

```
Test StdFunctional fnc {
    patburst = Global.burst;
    tim      = Global.tim;
    dclev    = Global.dclev;
    dccat    = Global.dccat;
    dcsel    = Global.selmeas;
}
```

```
Test Histogram hist {  
    test    = Global.fnc;  
}
```

Actions:

- Jim to rework Verigy example from STIL paper so that the STIL translation is consistent with its original Verigy code.

Reference documents (If logged into your google account, can edit. If not, can only view.)

- <http://spreadsheets.google.com/ccc?key=0AoKiPr1I9LY9dF95dkhSTVVqOU5GbWJyWFNhY0JPX0E&hl=en>
- Namespace resolution examples document:
<http://docs.google.com/Doc?docid=0AYKiPr1I9LY9ZGY4dmNjNTNfMGZkOGJ2bmZy&hl=en>
- Scratchpad spreadsheet: <https://spreadsheets0.google.com/ccc?key=tQ93VDnAZ-CI9RFKpPrPDzw&authkey=COzyro8K&hl=en&authkey=COzyro8K#gid=0>
- Scratchpad "Word" doc: https://docs1.google.com/document/d/1zVu2M8nTJsrm0nFbBhiuM8-YRt4ErYqdy_uSa3x3_T4/edit?authkey=CLrgwrsG#

Next meeting: 07/06/11

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>