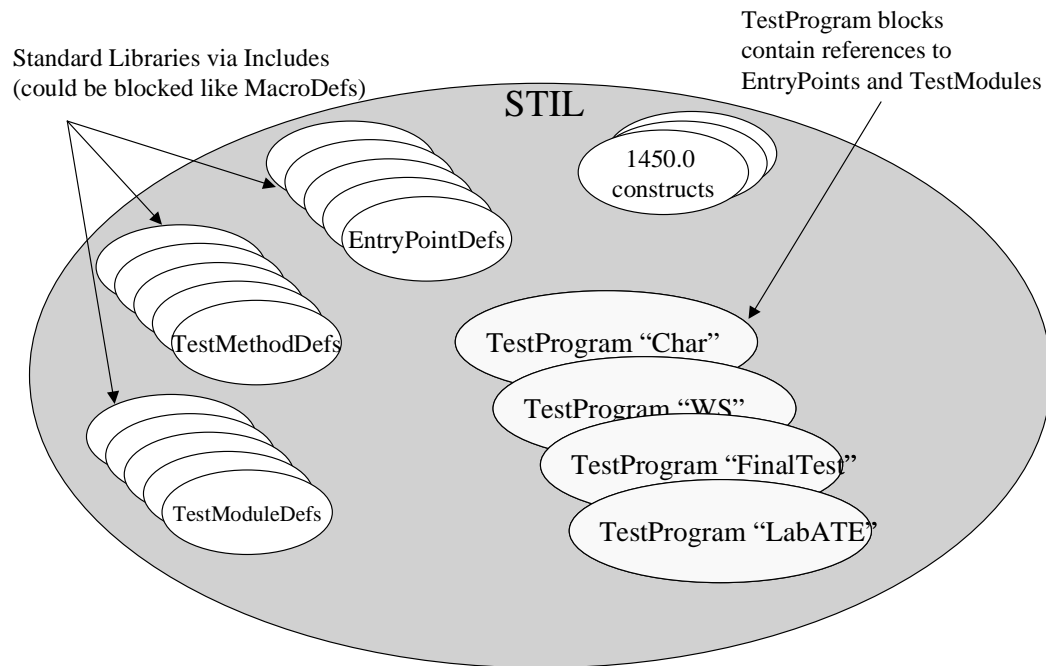


## 1.0 Top-Down Conceptual View

The P1450.4 constructs will reside within a IEEE 1450-1999 STIL Std file. Its top level constructs will add to the STIL top level constructs/keywords. Figure1 shows the TestProgram block, TestModuleDefs, TestMethodDef and EntryPointDefs blocks. The “Defs” blocks contain all definitions of their respective definition blocks and parallel the Macro-Def block in the 1450.0 STIL standard construct.

**FIGURE 1. Top Level Test Program Flow Constructs Diagram**



## 2.0 Test Program Flow Extension Terms

### 2.1 TestProgram Block:

The top level test program construct. There can be one or more. One may be global (unnamed). There may be one or more named TestProgram blocks in a STIL file.

### 2.2 Flow Block (or TestFlow):

This is the top level program flow construct. There can be one unnamed Flow block. There can be one or more named Flow blocks. This block contains definition of flow and bin entities that make up a given test program flow.

## **P1450.4 Test Program Flow Conceptual Model Discussion Document**

### **2.3 EntryPoint:**

This is a reference to a special program level task activated by the tester (tester operating system, system interrupts, etc.) This entry point references a TestModule. There is a general set of EntryPoint entities defined by this extension (i.e. OnStart, OnReset, etc.). These can be named and one instance of each can be unnamed and treated as global to any Flow block that does not declare a named one of each type. When a flow is active and a tester event requires an EntryPoint response, the associated EntryPoint TestModule/FlowModule that was declared, or the default if not declared is run.

### **2.4 FlowNode:**

(See Figure1) A node in the program flow that contains a ModuleRef (Body) that references a TestModule or FlowModule. This node has PreActions that defines the entry point into the node and may contain actions, declarations such as Spec/Category selection, etc. Absence of actions may in the Pre section may cause default actions (tbd). The Post section contains PostActions, Arbitrator, and ExitActions. The ExitActions give directives as to the follow-on flow path taken out of the FlowNode.

### **2.5 TestModule:**

(See Figures 2, 3 and 4)

### **2.6 BinNode and BinMap:**

Not yet defined/discussed

(Need two natures: terminal and flow-through)

### **2.7 TaskNode and DecisionNode:**

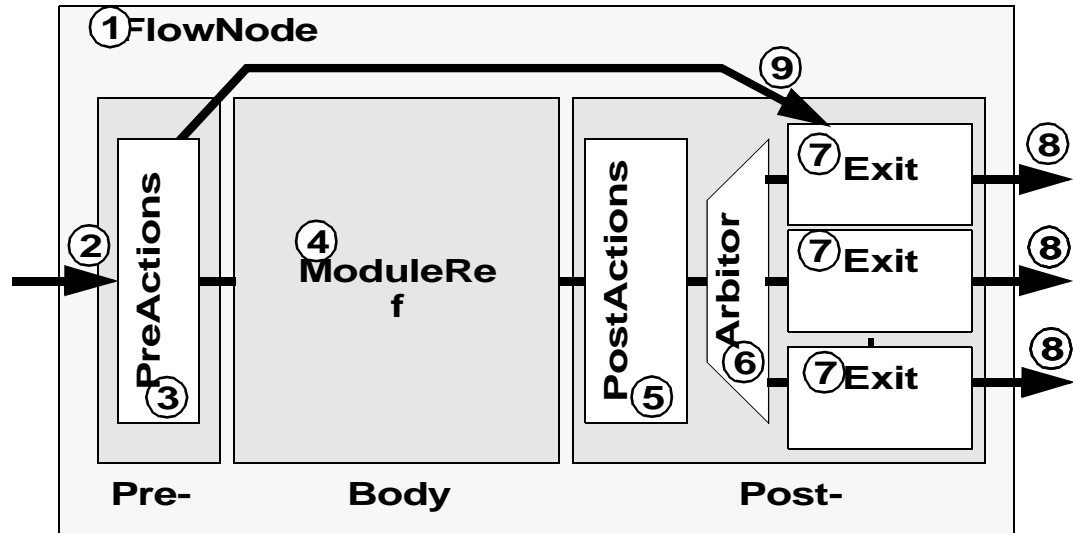
These are non-test type nodes used for non-test activities and flow decision content.

### **2.8 “Harness”**

This is an informative descriptive term that is not intended to have a keyword in the extension language. It is a common denominator descriptor for all TestModule instantiations.

### 3.0 FlowNode Conceptual Model

FIGURE 2. The FlowNode Conceptual Model Diagram (with its named components)



#### 3.1 FlowNode Components Descriptions

1. FlowNode
2. EntryPath
3. PreActions Block
4. ModuleRef (Module Reference)
5. PostActions Block
6. Arbitor Block
7. ExitActions Block
8. ExitPath
9. SkipPath (can goto to any ExitAction Block)

### **3.2 FlowNode Informative Term Descriptions**

1. “Pre-” portion
2. “Body” portion
3. “Post-” portion

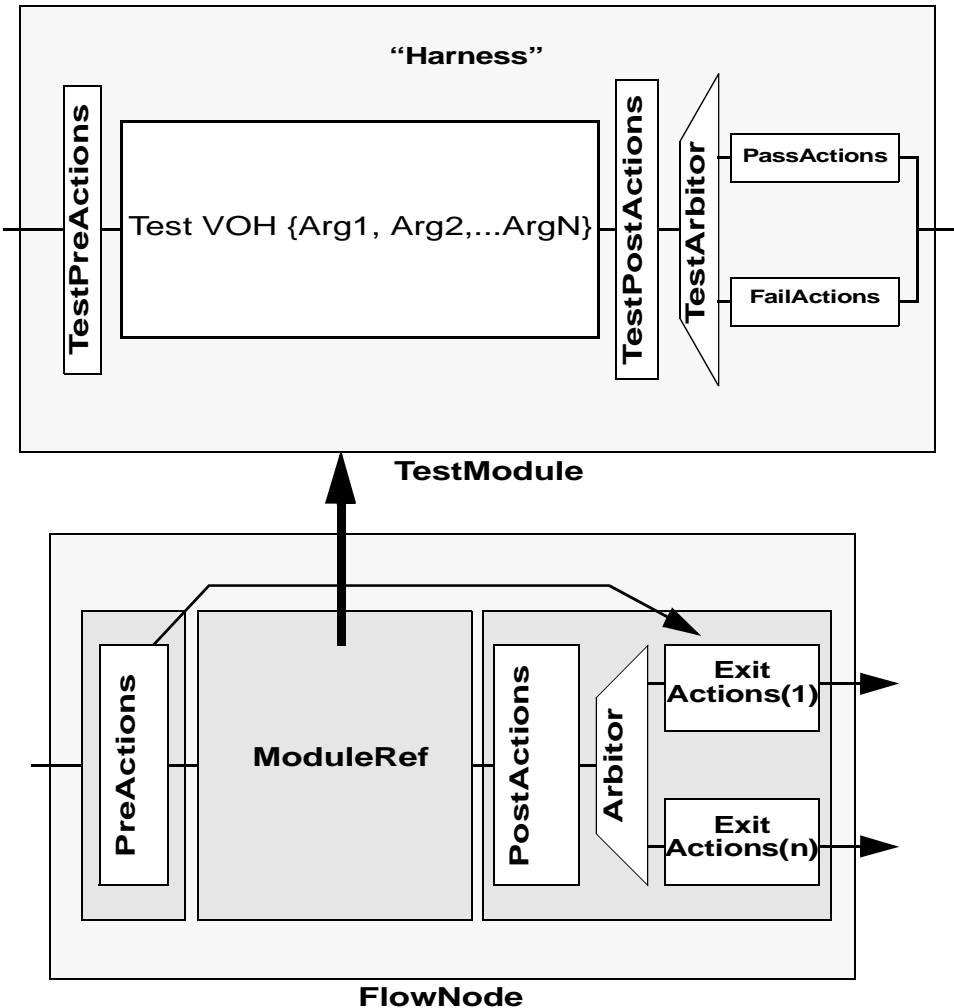
### **4.0 Relationship of FlowNodes to TestModules**

The instantiation of the TestMethod (i.e. VOH) can be “in-line” or “defined-before-use”. “Define-before-use” is the mechanism by which two or more FlowNodes can refer to the same “Test Object” (i.e. Module). Test Module instantiation involves placing the instantiated TestMethod inside the harness as shown to the left (the harness is the grey portion of the box labeled TestModule.)

TestMethod “VOH” { Arg1, Arg2,...ArgN } is the type definition, and

Test VOH { Arg1, Arg2,...ArgN } is the instantiation of the VOH TestMethod in the test module.

FIGURE 3. Block Diagram View of FlowNode Where ModuleRef References a TestModule

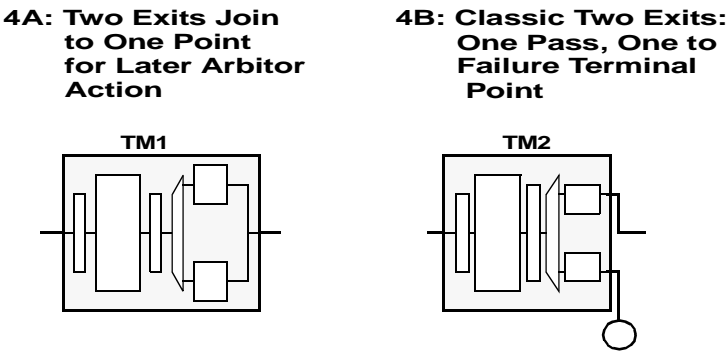


## 5.0 TestModule Characteristics

### 5.1 Two Types of “Outflow” Configurations for TestModules

P1450.4 Test Program Flow Conceptual Model Discussion Document

FIGURE 4. Conceptual Block Diagrams of the Two Outflow Types



5.2 TestFlow as the TestModule Body

FIGURE 5. TestModule Referencing a TestFlow

