Concurrent assertions in classes

Problem
Many users requested the availability of SVA in classes as they allow the verification of interface
variables in conjunction with class variables. This is particularly important in UVM for applications
such as scoreboards, monitors, and drivers.

Possible solutions
Assertions in the following manner:
   a) Assertions become active dynamically.
   b) Tool will have to start collecting and reporting on concurrent assertions in a class when the class
      is instanced.
   c) Tool will have to stop collecting and freeze the reporting on concurrent assertions in a class when
      the classes is nulled.
   d) A reference to a null object in an assertion would make that assertion vacuous.

Example

```verbatim
interface A_if (input logic clk);
// ...
clocking driver_cb @ (posedge clk);
    output rst_n, data_in, ld, kind_cp;
    input A;
endclocking : driver_cb

    modport drvr_if_mp (clocking driver_cb);
endinterface : A_if

class A_driver extends uvm_driver #(A_xactn, A_xactn);
    string tID;
    virtual interface A_if.drvr_if_mp vif;
    rand bit[3:0] ct;
    bit a, b, c;
    function new(string name, uvm_component parent);
        //...
        endfunction : new

        `uvm_component_utils_begin(A_driver)
        `uvm_field_object(req, UVM_ALL_ON)
        `uvm_component_utils_end

        task get_and_drive();
            ap_abc: assert property(@(this.vif.driver_cb)
                this.vif.A_cb.ld |=> a ##1 ct <= 4'b1100);
        endtask : get_and_drive

endclass : A_driver

module forsv_ac;
    A_driver a_dvr1;
    initial begin
        a_dvr1=new(...);
        // assertions for a_dvr1 become active ...
        // ...spme operations with cycles
        a_dvr1=null;
        // assertions for a_dvr1 stop collecting
    end
endmodule
```