2026 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SANTA CLARA, CA, USA
MARCH 2 - 5, 2026

PARADIGM WORKS
Verification Excellence

Visualizing SystemVerilog and UVM

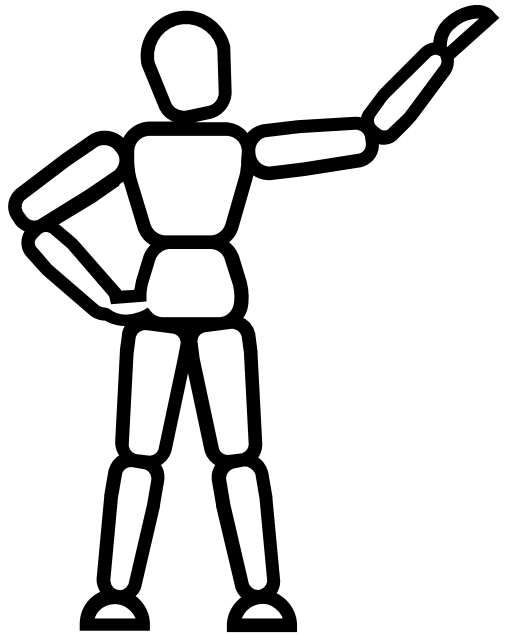
Jamie Ridgeway
Paradigm-Works, Inc.

PARADIGM® WORKS
Verification Excellence

Agenda

- Why and what?
- How?

Somone is new to the verification team



- How do *you* get them up to speed?
- What documents are available:

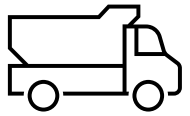
✓ Hardware specs?

✓ Verification plan?

✗ Testbench doc?

Testbench Documentation? Ha!

- Who has time for this?
- Always out of date
- Never represents actual implementation
- Becomes essentially worthless to the project



Motivation

Learning Styles

- ✓ Visual
- ✓ Auditory
  - Reading/Writing
  - Kinesthetic (doing)

Conference Room

- Whiteboard drawings
- Talking about the project

Motivation

Learning Styles

- Visual
- Auditory
- Reading/Writing
- Kinesthetic (doing)

Conference Room

- Whiteboard drawings
- Talking about the project

Individual

- Specs / reference
- Coding / simulation

Three tools | Two goals

- svpre.pl *preprocess*
- svdiag.pl *class diagram*
- uvmtopo.pl *testbench diagram*

- Understanding
- Documentation

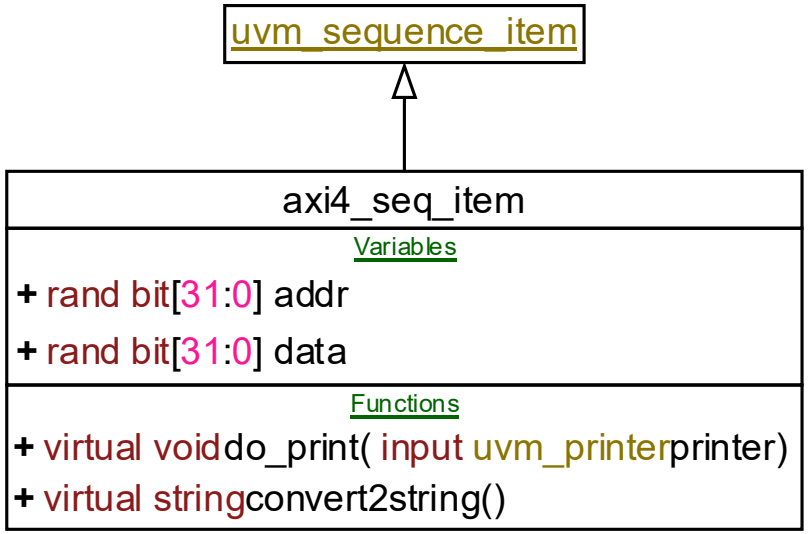
svdiag.pl

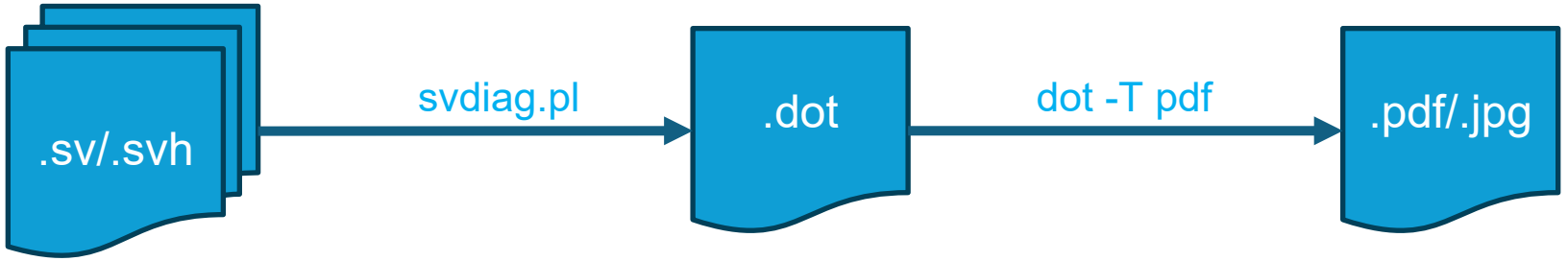
- Class inheritance
- Class contents
- Generates DOT diagrams

```
1  class axi4_seq_item extends uvm_sequence_item;
2      `uvm_object_utils(axi4_seq_item)
3
4      // Variable: addr
5      // Address targeted for reading or writing.
6      rand bit[31:0] addr, data;
7
8      // Function: do_print
9      // UVM compatible printing.
10     extern virtual function void do_print(uvm_printer printer);
11     extern virtual function string convert2string();
12
13 endclass
```

svdiag.pl

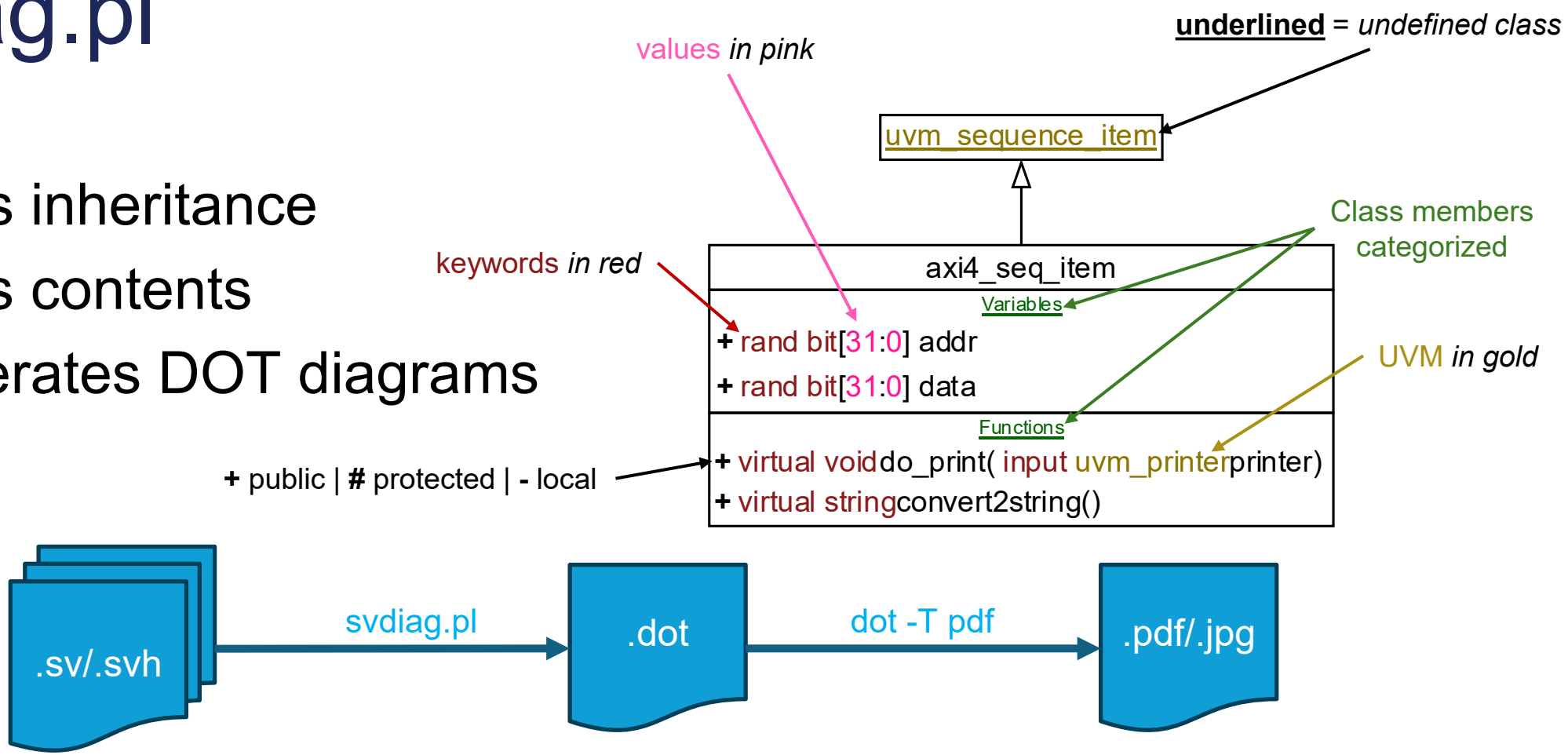
- Class inheritance
- Class contents
- Generates DOT diagrams





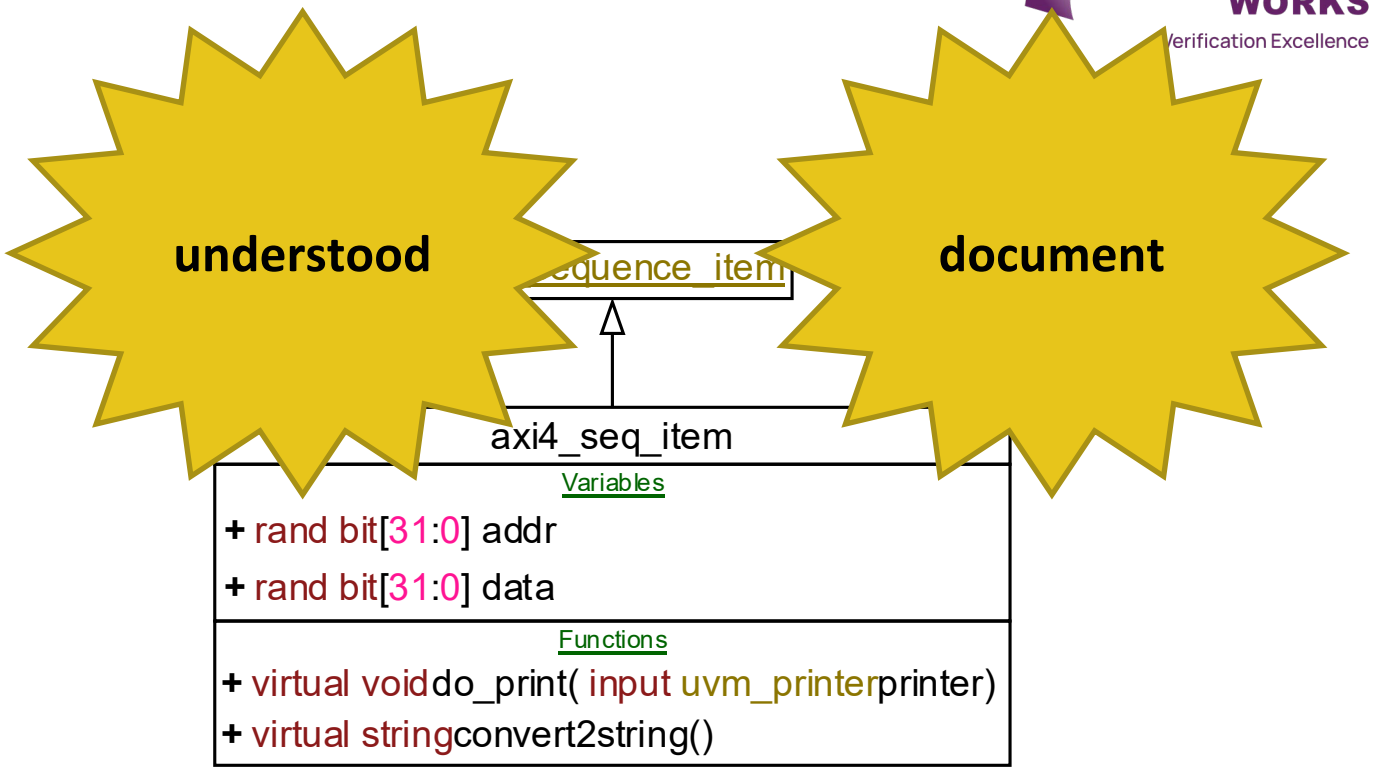
svdiag.pl

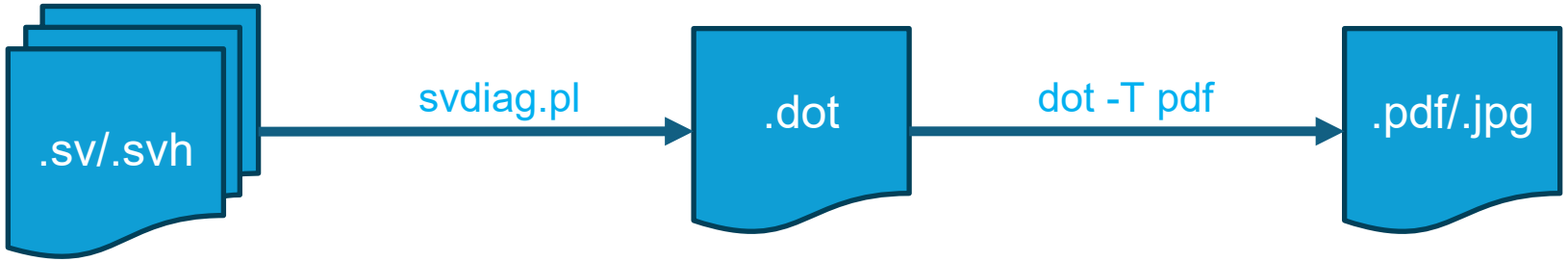
- Class inheritance
- Class contents
- Generates DOT diagrams



svdiag.pl

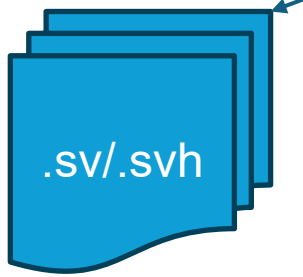
- Class inheritance
- Class contents
- Generates DOT diagrams





svpre.pl

- Preprocesses SystemVerilog
  - `include “file”
  - `ifdef-`elsif-`else-`endif
  - `define macro_name …
  - `macro_name instantiation




```
1 `include "uvm_macros.svh"
2 `include "uvm_sequence_item.svh"
3 `include "axi4_seq_item.sv"
```

`including our sequence item

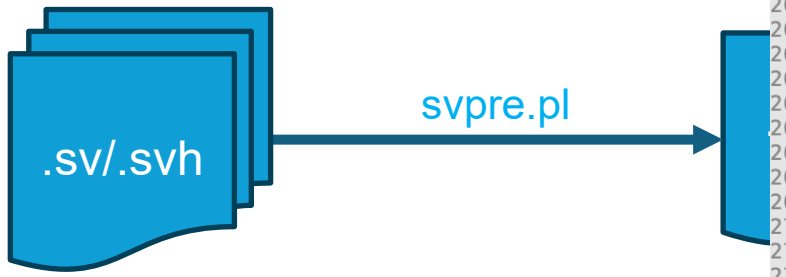
svpre.pl

- Preprocesses SystemVerilog
  - `include “file”
  - `ifdef-`elsif-`else-`endif
  - `define macro_name …
  - `macro_name instantiation

```
1 `include "uvm_macros.svh"
2 `include "uvm_sequence_item.svh"
3 `include "axi4_seq_item.sv"
```

svpre.pl

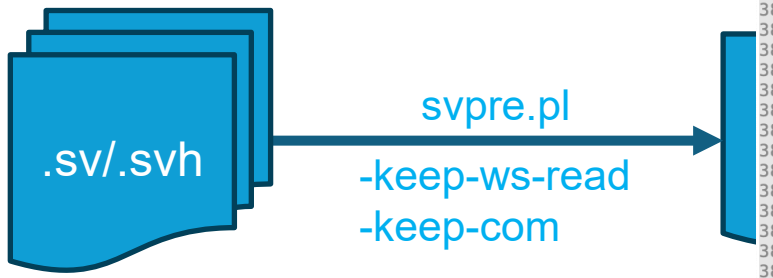
- Preprocesses SystemVerilog
  - `include “file”
  - `ifdef-`elsif-`else-`endif
  - `define macro_name …
  - `macro_name instantiation




```
231 class axi4_seq_item extends uvm_sequence_item ;
232 typedef uvm_object_registry # ( axi4_seq_item , "axi4_seq_item" ) type_id ;
233 static function type_id get_type ( ) ;
234 return type_id :: get ( ) ;
235 endfunction
236 virtual function uvm_object_wrapper get_object_type ( ) ;
237 return type_id :: get ( ) ;
238 endfunction
239 function uvm_object create ( string name = "" ) ;
240 axi4_seq_item tmp ;
241 if ( name == "" ) tmp = new ( ) ;
242 else tmp = new ( name ) ;
243 return tmp ;
244 endfunction
245 const static string type_name = "axi4_seq_item" ;
246 virtual function string get_type_name ( ) ;
247 return type_name ;
248 endfunction
249 function void __m_uvm_field_automation ( uvm_object tmp_data__ ,
250 int what__ ,
251 string str__ ) ;
252 begin
253 axi4_seq_item local_data__ ;
254 typedef axi4_seq_item ___local_type____ ;
255 string string_aa_key ;
256 uvm_object __current_scopes [ $ ] ;
257 if ( what__ inside { UVM_SETINT , UVM_SETSTR , UVM_SETOBJ } ) begin
258 if ( __m_uvm_status_container . m_do_cycle_check ( this ) ) begin
259 return ;
260 end
261 else
262 __current_scopes = __m_uvm_status_container . m_uvm_cycle_scopes ;
263 end
264 super . __m_uvm_field_automation ( tmp_data__ , what__ , str__ ) ;
265 if ( tmp_data__ != null )
266 if ( ! $cast ( local_data__ , tmp_data__ ) ) return ;
267 end
268 endfunction
269 rand bit [ 31 : 0 ] addr , data ;
270 extern virtual function void do_print ( uvm_printer printer ) ;
271 extern virtual function string convert2string ( ) ;
272 endclass
```

svpre.pl

- Preprocesses SystemVerilog
  - `include “file”
  - `ifdef-`elsif-`else-`endif
  - `define macro_name …
  - `macro_name instantiation




```
3833 class axi4_seq_item extends uvm_sequence_item ;
3834
3835
3836
3837   typedef uvm_object_registry # ( axi4_seq_item , "axi4_seq_item" ) type_id ;
3838   static function type_id get_type ( ) ;
3839     return type_id :: get ( ) ;
3840   endfunction
3841   virtual function uvm_object_wrapper get_object_type ( ) ;
3842     return type_id :: get ( ) ;
3843      endfunction
3844
3845   function uvm_object create ( string name = "" ) ;
3846     axi4_seq_item tmp ;
3847     if ( name == "" ) tmp = new ( ) ;
3848     else tmp = new ( name ) ;
3849     return tmp ;
3850      endfunction
3851
3852   const static string type_name = "axi4_seq_item" ;
3853   virtual function string get_type_name ( ) ;
3854     return type_name ;
3855     endfunction
3856
3857   function void __m_uvm_field_automation ( uvm_object tmp_data__ ,
3858                                            int what__ ,
3859                                            string str__ ) ;
3860   begin
3861     axi4_seq_item local_data__ ;
3862     typedef axi4_seq_item ___local_type____ ;
3863     string string_aa_key ;
3864     uvm_object __current_scopes [ $ ] ;
3865     if ( what__ inside { UVM_SETINT , UVM_SETSTR , UVM_SETOBJ } ) begin
3866        if ( __m_uvm_status_container . m_do_cycle_check ( this ) ) begin
3867           return ;
3868        end
3869        else
3870           __current_scopes = __m_uvm_status_container . m_uvm_cycle_scopes ;
3871     end
3872     super . __m_uvm_field_automation ( tmp_data__ , what__ , str__ ) ;
3873     if ( tmp_data__ != null )
3874           if ( ! $cast ( local_data__ , tmp_data__ ) ) return ;
3875
3876     end
3877   endfunction
3878
3879    // Variable: addr
3880    // Address targeted for reading or writing.
3881    rand bit [ 31 : 0 ] addr , data ;
3882
3883    // Function: do_print
3884    // UVM compatible printing.
3885    extern virtual function void do_print ( uvm_printer printer ) ;
3886    extern virtual function string convert2string ( ) ;
3887
3888 endclass
```

whitespace is preserved

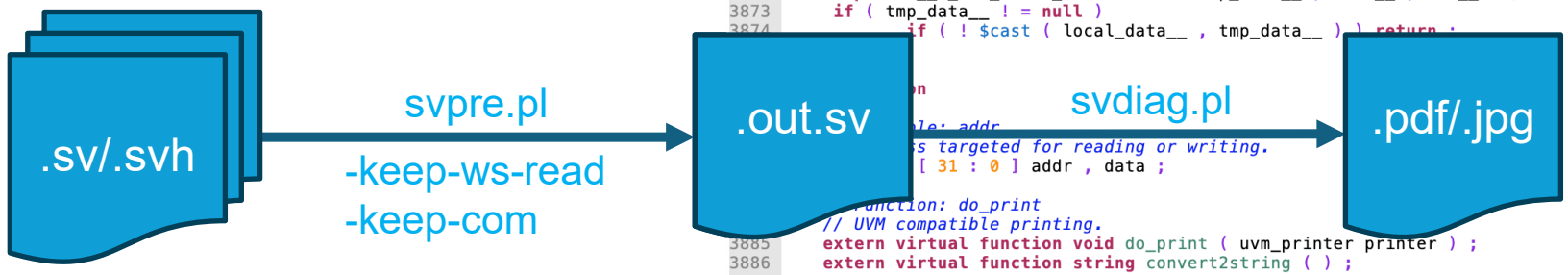
comments are preserved

svpre.pl

- Preprocesses SystemVerilog
  - `include “file”
  - `ifdef-`elsif-`else-`endif
  - `define macro_name …
  - `macro_name instantiation

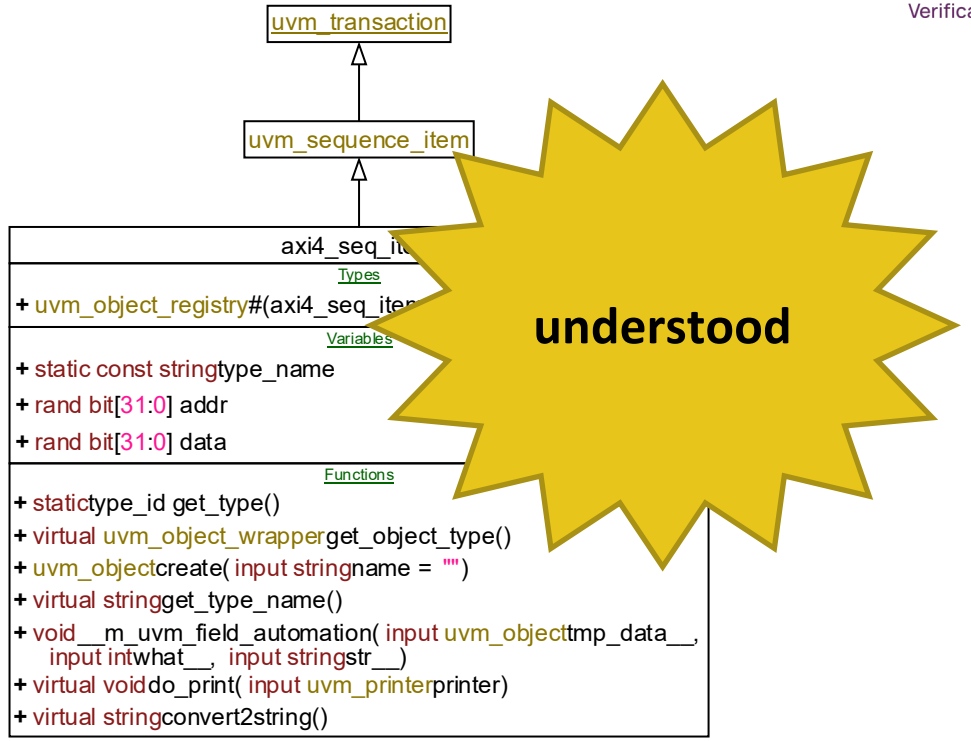
```
3833 class axi4_seq_item extends uvm_sequence_item ;
3834
3835
3836
3837    typedef uvm_object_registry # ( axi4_seq_item , "axi4_seq_item" ) type_id ;
3838    static function type_id get_type ( ) ;
3839      return type_id :: get ( ) ;
3840    endfunction
3841    virtual function uvm_object_wrapper get_object_type ( ) ;
3842      return type_id :: get ( ) ;
3843       endfunction
3844
3845    function uvm_object create ( string name = "" ) ;
3846      axi4_seq_item tmp ;
3847      if ( name == "" ) tmp = new ( ) ;
3848      else tmp = new ( name ) ;
3849      return tmp ;
3850       endfunction
3851
3852    const static string type_name = "axi4_seq_item
3853    virtual function string get_type_name ( ) ;
3854      return type_name ;
3855       endfunction
3856
3857    function void __m_uvm_field_automation ( uvm_object
3858                                             int what__ ,
3859                                             string str__ ) ;
3860    begin
3861      axi4_seq_item local_data__ ;
3862      typedef axi4_seq_item ___local_type____ ;
3863      string string_aa_key ;
3864      uvm_object __current_scopes [ $ ] ;
3865      if ( what__ inside { UVM_SETINT , UVM_SETSTR , UVM_SETOBJ } ) begin
3866         if ( __m_uvm_status_container . m_do_cycle_check ( this ) ) begin
3867             return ;
3868         end
3869         else
3870             __current_scopes = __m_uvm_status_container . m_uvm_cycle_scopes ;
3871      end
3872      super . __m_uvm_field_automation ( tmp_data__ , what__ , str__ ) ;
3873      if ( tmp_data__ != null )
3874          if ( ! $cast ( local_data__ , tmp_data__ ) ) return ;
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884     // UVM compatible printing.
3885     extern virtual function void do_print ( uvm_printer printer ) ;
3886     extern virtual function string convert2string ( ) ;
3887
3888 endclass
```

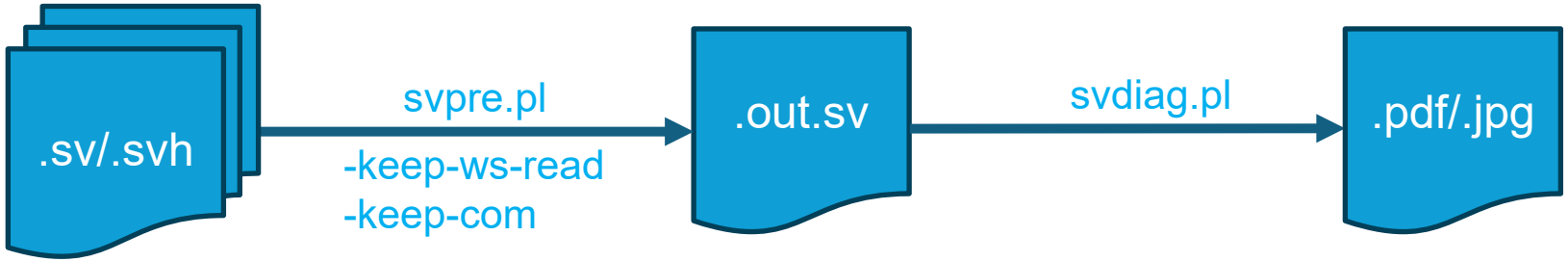
understood



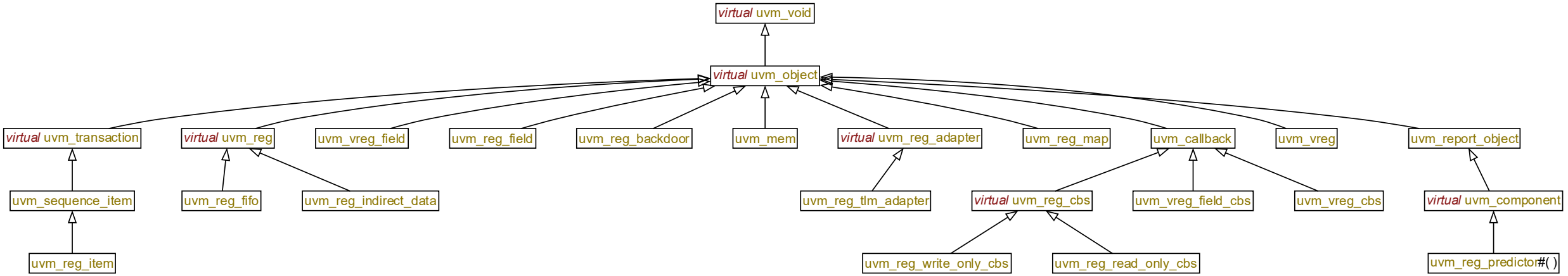
svpre.pl

- Preprocesses SystemVerilog
  - `include “file”
  - `ifdef-`elsif-`else-`endif
  - `define macro_name …
  - `macro_name instantiation





UVM REG Related Classes

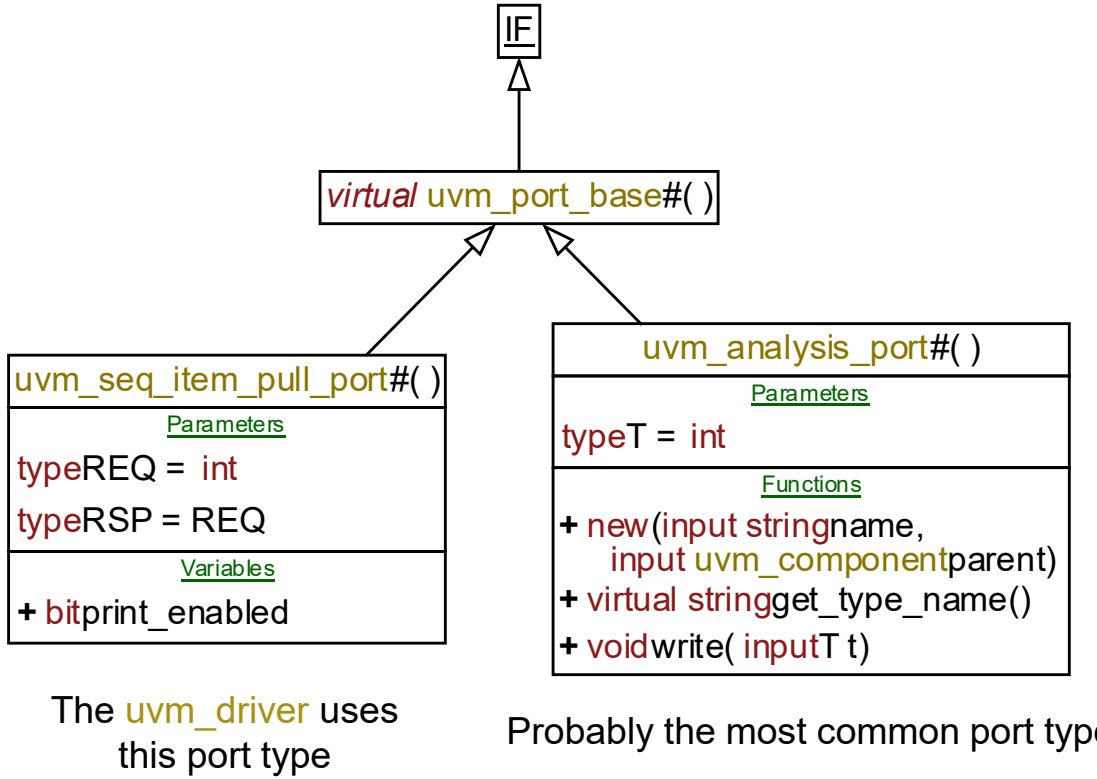


focuses output on specified classes and their inheritance lineage

no members are shown

svdiag.pl **-class** uvm_mem,uvm_reg,… **-hide-detail** …

UVM Driver's Port & UVM Analysis Port



The uvm_driver uses this port type

Probably the most common port type

svdiag.pl -class uvm_seq_item_pull_port,uvm_analysis_port -hide-lineage

Testbench Documentation

- Develop architecture diagrams
  - Explain what they are intended to do
  - Explain how they interact

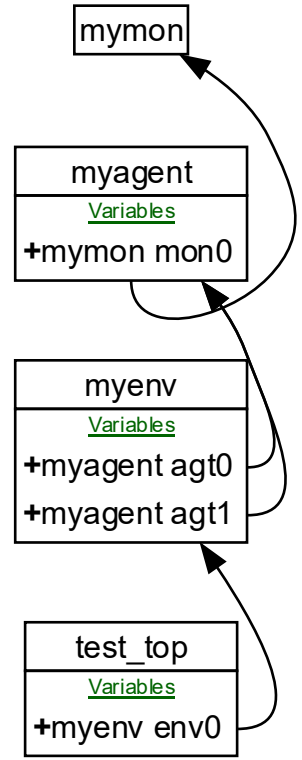
- Develop block / class diagrams  svdiag.pl
  - Highlight only what is important
  - Use access protections and show only public

uvmtopo.pl

- Can *you* read the UVM topology printout?

```
# UVM_INFO @ 0.000ns: reporter [UVMTOP] UVM testbench topology:
# ----------------------------------------
# Name             Type        Size  Value
# ----------------------------------------
# uvm_test_top     test_top    -     @1
#  env0            myenv       -     @2
#    agt0          myagent     -     @3
#    agt1          myagent     -     @3
#      mon0        mymon       -     @3
# ----------------------------------------
```

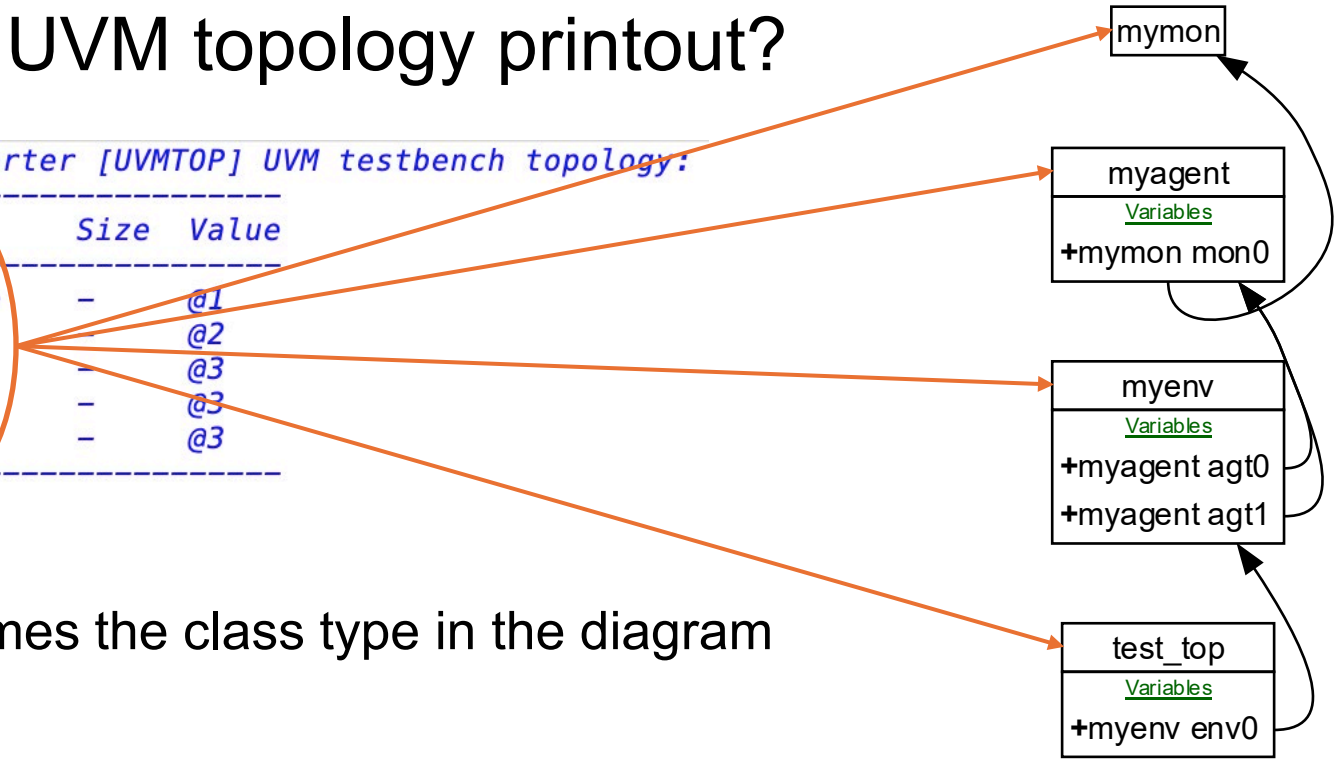
uvmtopo.pl



uvmtopo.pl

- Can *you* read the UVM topology printout?

```
1  # UVM_INFO @ 0.000ns: reporter [UVMTOP] UVM testbench topology:
2  # ----------------------------------------------
3  # Name              Type          Size  Value
4  # ----------------------------------------------
5  # uvm_test_top      test_top      -     @1
6  #   env0            myenv         -     @2
7  #     agt0          myagent       -     @3
8  #     agt1          myagent       -     @3
9  #       mon0        mymon         -     @3
10 # ----------------------------------------------
```



The Type column becomes the class type in the diagram

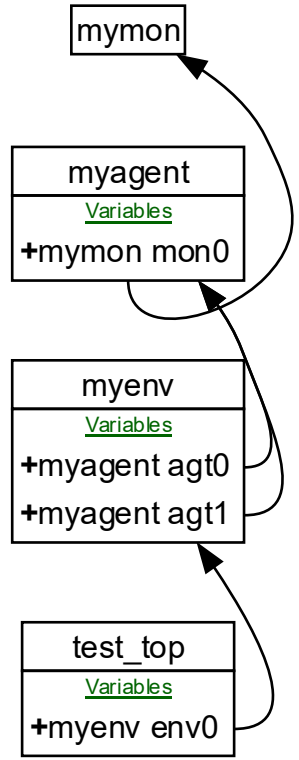
uvmtopo.pl

- Can *you* read the UVM topology printout?

```
# UVM_INFO @ 0.000ns: reporter [UVMTOP] UVM testbench topology:
# ----------------------------------------------------
# Name                Type          Size  Value
# ----------------------------------------------------
# uvm_test_top        test_top      -     @1
#   env0              myenv         -     @2
#     agt0            myagent       -     @3
#     agt1            myagent       -     @3
#       mon0          mymon         -     @3
# ----------------------------------------------------
```

The type comes from get_type_name()

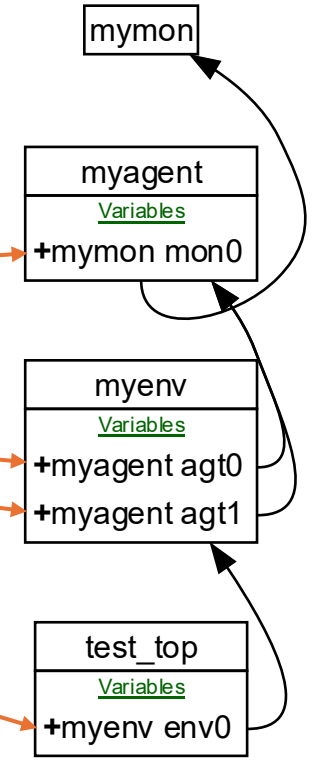
Parameterized classes do not define this function in the uvm macro



uvmtopo.pl

- Can *you* read the UVM topology printout?

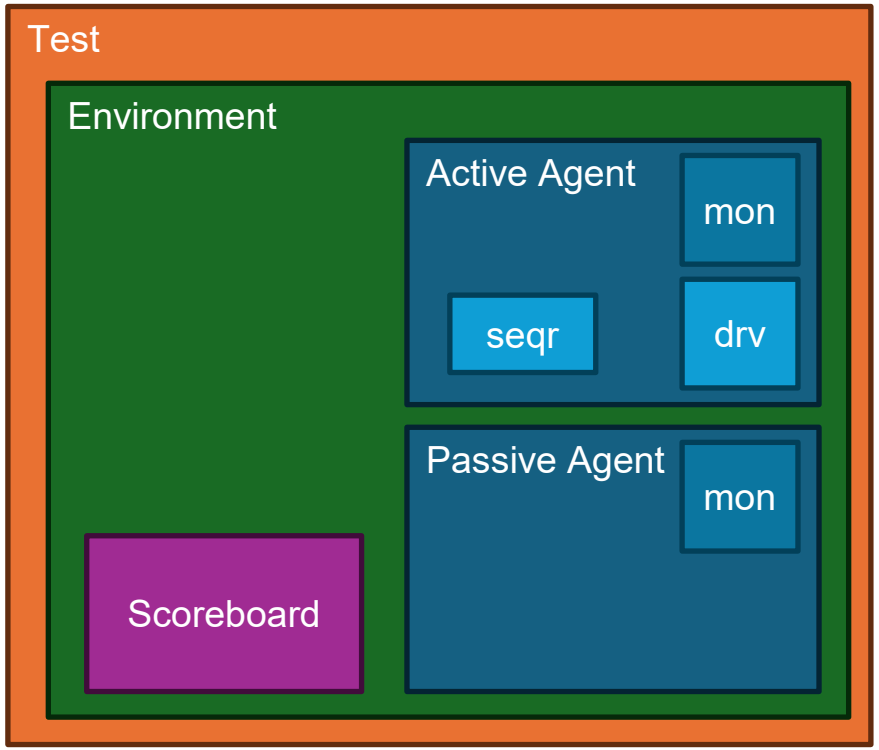
```
# UVM_INFO @ 0.000ns: reporter [UVMTOP] UVM testbench topology:
# ----------------------------------------------------------
# Name              Type        Size  Value
# ----------------------------------------------------------
# uvm_test_top      test_top    -     @1
#   env0            myenv       -     @2
#     agt0          myagent     -     @3
#     agt1          myagent     -     @3
#       mon0        mymon       -     @3
# ----------------------------------------------------------
```



The Name column defines hierarchy and class members

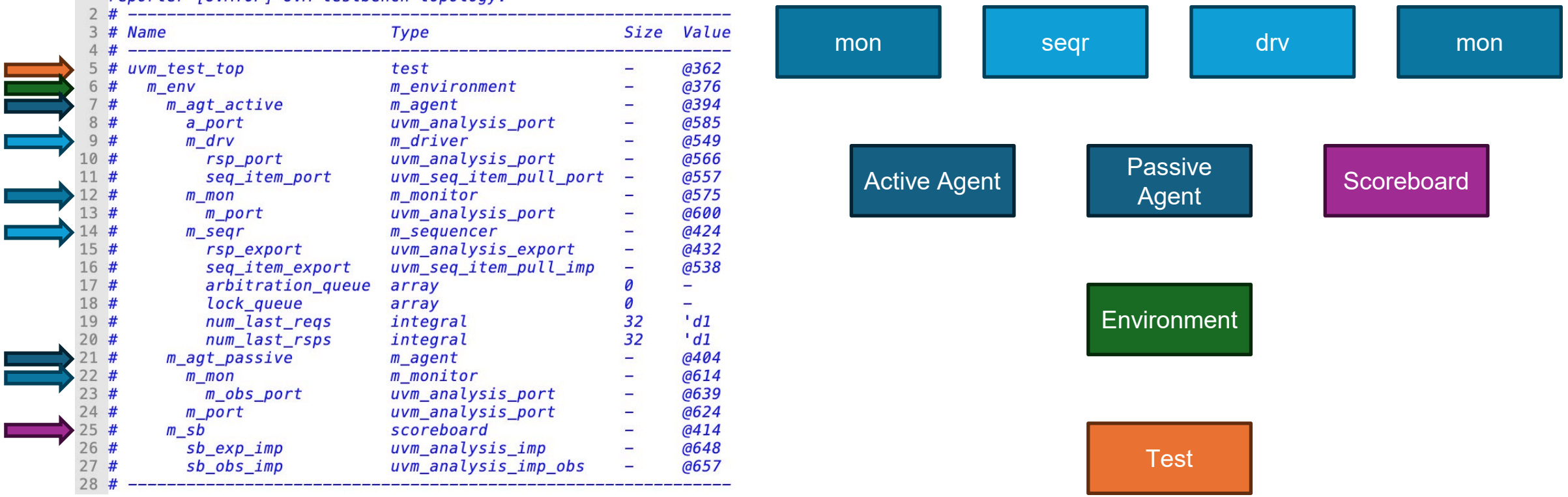
Standard UVM Testbench Style

```
# UVM_INFO verilog_src/uvm-1.2/src/base/uvm_root.svh(579) @ 0:
reporter [UVMTOP] UVM testbench topology:
# ----------------------------------------------------------------
# Name                        Type                      Size  Value
# ----------------------------------------------------------------
# uvm_test_top                test                      -     @362
#   m_env                     m_environment             -     @376
#     m_agt_active            m_agent                   -     @394
#       a_port                uvm_analysis_port         -     @585
#       m_drv                 m_driver                  -     @549
#         rsp_port            uvm_analysis_port         -     @566
#         seq_item_port       uvm_seq_item_pull_port    -     @557
#       m_mon                 m_monitor                 -     @575
#         m_port              uvm_analysis_port         -     @600
#       m_seqr                m_sequencer               -     @424
#         rsp_export          uvm_analysis_export       -     @432
#         seq_item_export     uvm_seq_item_pull_imp     -     @538
#         arbitration_queue   array                     0     -
#         lock_queue          array                     0     -
#         num_last_reqs       integral                  32    'd1
#         num_last_rsps       integral                  32    'd1
#     m_agt_passive           m_agent                   -     @404
#       m_mon                 m_monitor                 -     @614
#         m_obs_port          uvm_analysis_port         -     @639
#       m_port                uvm_analysis_port         -     @624
#     m_sb                    scoreboard                -     @414
#       sb_exp_imp            uvm_analysis_imp          -     @648
#       sb_obs_imp            uvm_analysis_imp_obs      -     @657
# ----------------------------------------------------------------
```

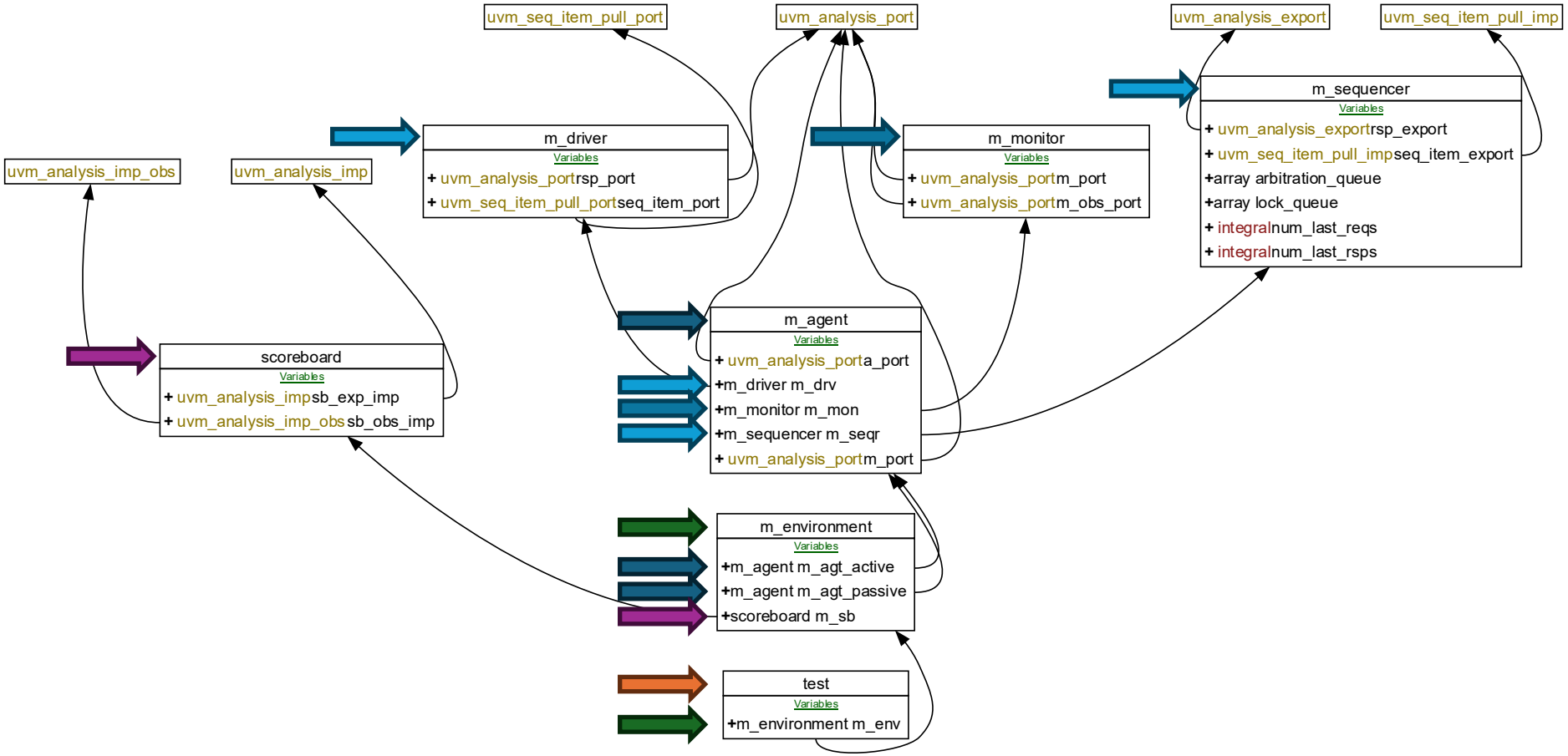


Standard UVM Testbench Style

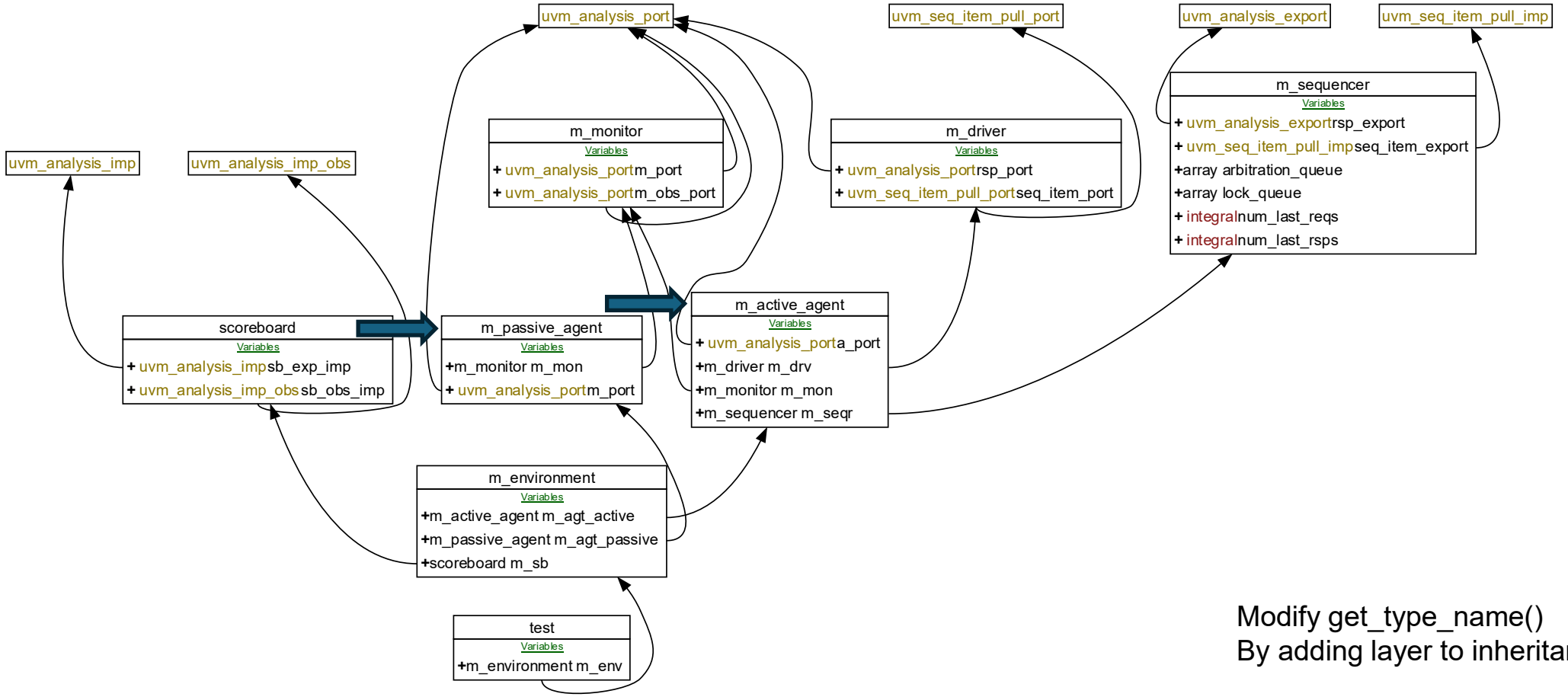
```
# UVM_INFO verilog_src/uvm-1.2/src/base/uvm_root.svh(579) @ 0:
reporter [UVMTOP] UVM testbench topology:
# ----------------------------------------------------------------
# Name                        Type                      Size  Value
# ----------------------------------------------------------------
# uvm_test_top                test                      -     @362
#   m_env                     m_environment             -     @376
#     m_agt_active            m_agent                   -     @394
#       a_port                uvm_analysis_port         -     @585
#       m_drv                 m_driver                  -     @549
#         rsp_port            uvm_analysis_port         -     @566
#         seq_item_port       uvm_seq_item_pull_port    -     @557
#       m_mon                 m_monitor                 -     @575
#         m_port              uvm_analysis_port         -     @600
#       m_seqr                m_sequencer               -     @424
#         rsp_export          uvm_analysis_export       -     @432
#         seq_item_export     uvm_seq_item_pull_imp     -     @538
#         arbitration_queue   array                     0     -
#         lock_queue          array                     0     -
#         num_last_reqs       integral                  32    'd1
#         num_last_rsps       integral                  32    'd1
#     m_agt_passive           m_agent                   -     @404
#       m_mon                 m_monitor                 -     @614
#         m_obs_port          uvm_analysis_port         -     @639
#       m_port                uvm_analysis_port         -     @624
#     m_sb                    scoreboard                -     @414
#       sb_exp_imp            uvm_analysis_imp          -     @648
#       sb_obs_imp            uvm_analysis_imp_obs      -     @657
# ----------------------------------------------------------------
```



Standard UVM Testbench Style



Split Agent Insts into Different Types



Modify get_type_name()
By adding layer to inheritance?

Limitations

- Interface classes are not supported

```
class A extends B implements C;
interface class C;
```

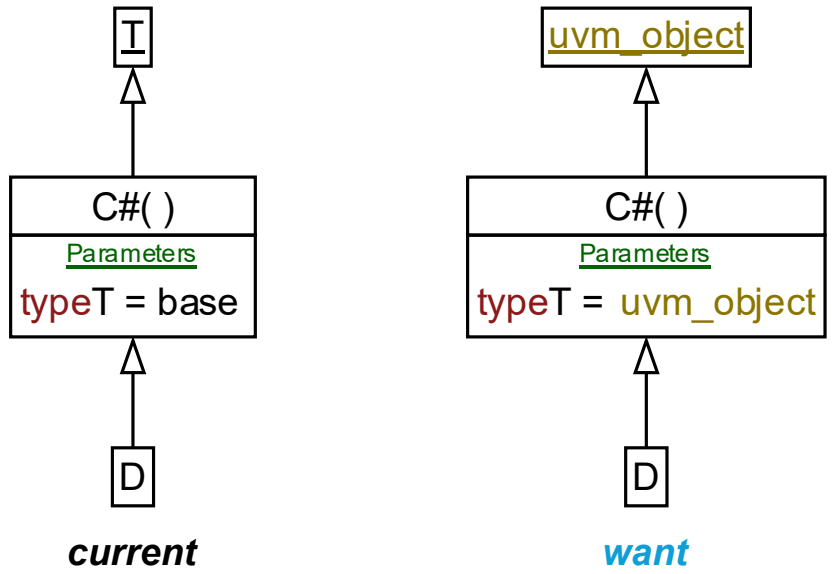
Limitations

- Interface classes are not supported
- Packages are ignored
- Global / package level functions are ignored
- Type parameters are not resolved
- Opaque type parameter extensions are not resolved

Limitations

- Type parameters are not resolved
- Opaque type parameter extensions are not resolved

```
class C #(type T = base) extends T;
class D extends C#(uvm_object);
```



Limitations

- uvmtopo.pl has a lot of opportunity for improvement

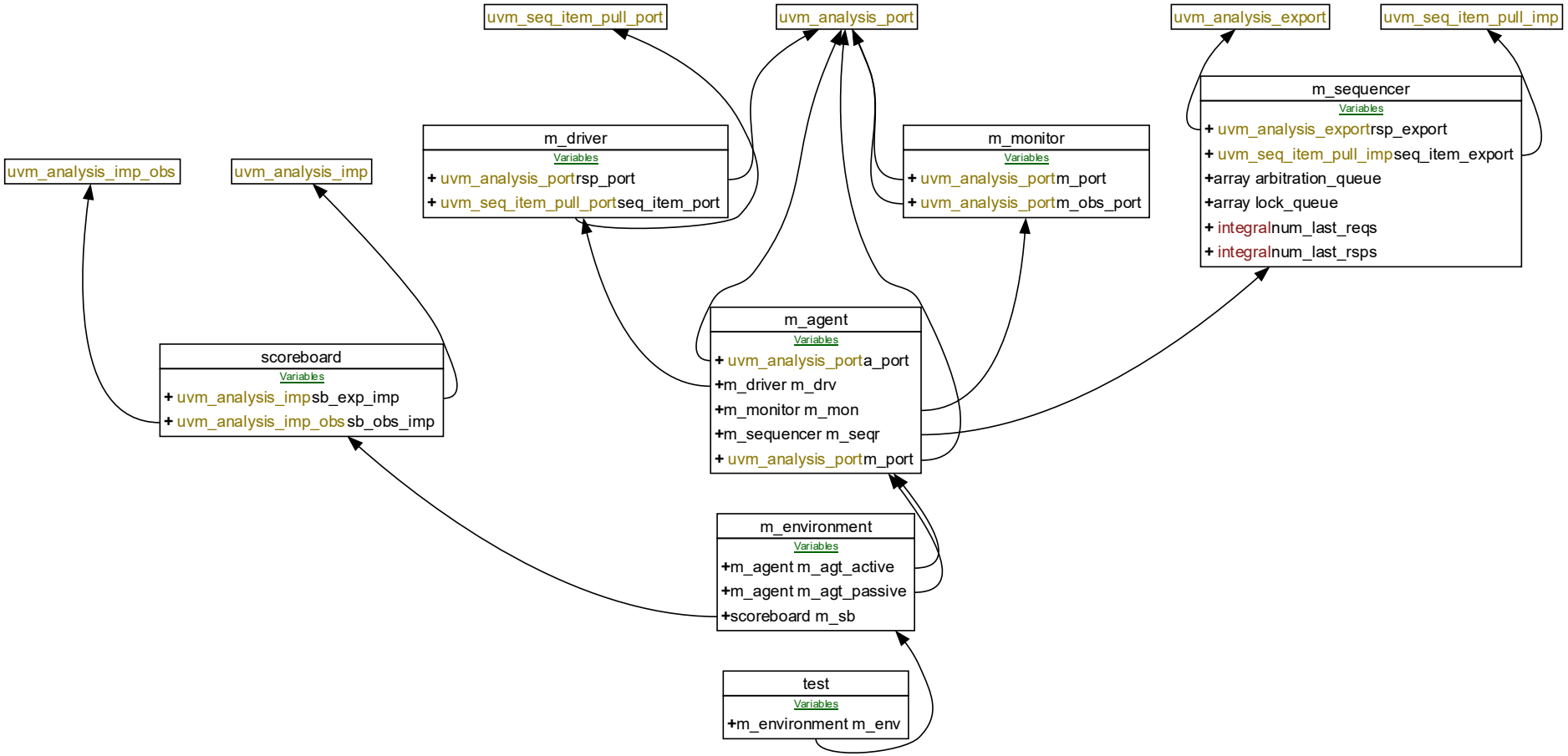
Conclusion

- I don’t if this can help testbench documentation
- I do hope tools like these can help a new team member come up to speed faster
- Freeware GPL License
- https://github.com/svloghack/svdiag

Extra / Supporting

If necessary, during discussion

Standard UVM Testbench Style



Splitting Agent Types

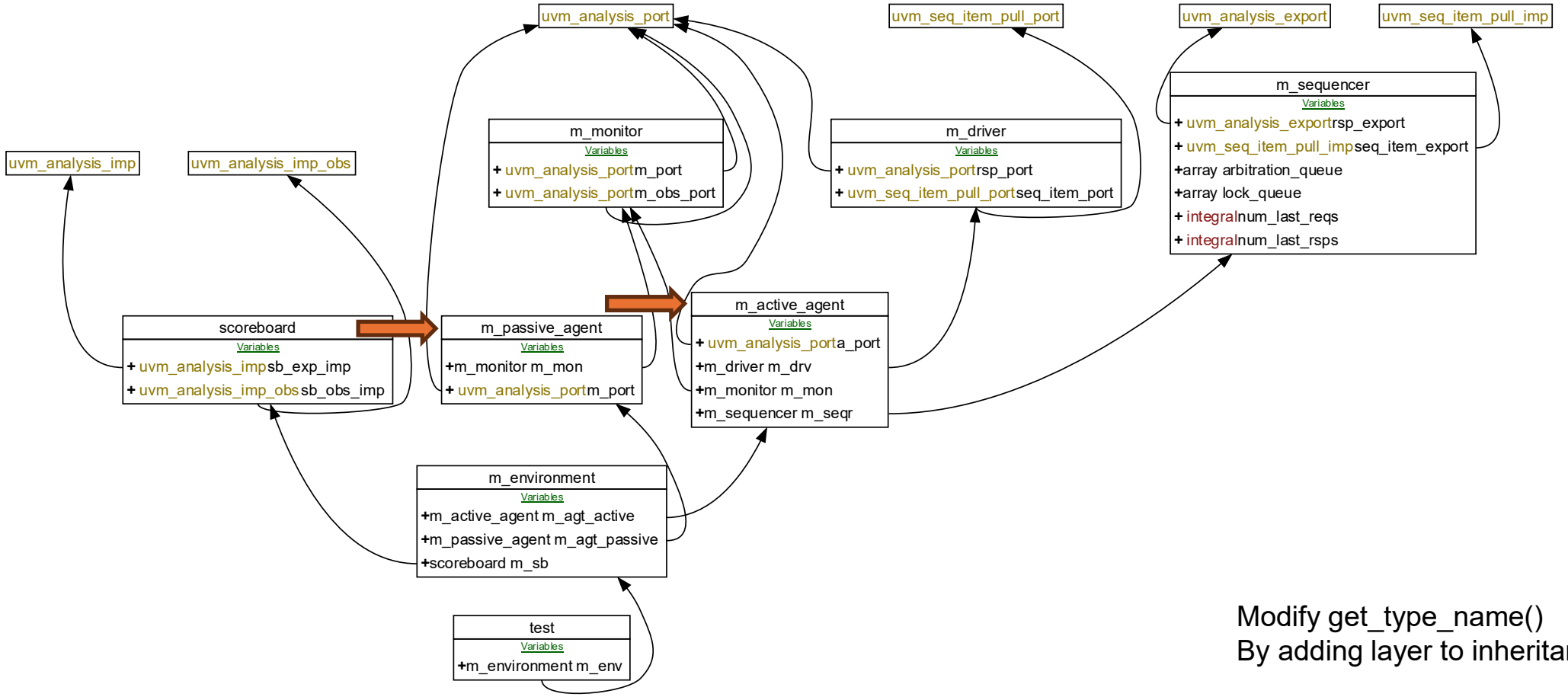
```
# UVM_INFO verilog_src/uvm-1.2/src/base/uvm_root.svh(579) @ 0:
reporter [UVMTOP] UVM testbench topology:
# ------------------------------------------------------------
# Name                       Type                    Size  Value
# ------------------------------------------------------------
# uvm_test_top               test                    -     @362
#   m_env                    m_environment           -     @376
#     m_agt_active           m_agent                 -     @394
#       a_port               uvm_analysis_port       -     @585
#       m_drv                m_driver                -     @549
#         rsp_port           uvm_analysis_port       -     @566
#         seq_item_port      uvm_seq_item_pull_port  -     @557
#       m_mon                m_monitor               -     @575
#         m_port             uvm_analysis_port       -     @600
#       m_seqr               m_sequencer             -     @424
#         rsp_export         uvm_analysis_export     -     @432
#         seq_item_export    uvm_seq_item_pull_imp   -     @538
#         arbitration_queue  array                   0     -
#         lock_queue         array                   0     -
#         num_last_reqs      integral                32    'd1
#         num_last_rsps      integral                32    'd1
#     m_agt_passive          m_agent                 -     @404
#       m_mon                m_monitor               -     @614
#         m_obs_port         uvm_analysis_port       -     @639
#       m_port               uvm_analysis_port       -     @624
#     m_sb                   scoreboard              -     @414
#       sb_exp_imp           uvm_analysis_imp        -     @648
#       sb_obs_imp           uvm_analysis_imp_obs    -     @657
# ------------------------------------------------------------
```

Active/Passive agents are same type

```
# UVM_INFO verilog_src/uvm-1.2/src/base/uvm_root.svh(579) @ 0:
reporter [UVMTOP] UVM testbench topology:
# ------------------------------------------------------------
# Name                       Type                    Size  Value
# ------------------------------------------------------------
# uvm_test_top               test                    -     @362
#   m_env                    m_environment           -     @376
#     m_agt_active           m_active_agent          -     @394
#       a_port               uvm_analysis_port       -     @585
#       m_drv                m_driver                -     @549
#         rsp_port           uvm_analysis_port       -     @566
#         seq_item_port      uvm_seq_item_pull_port  -     @557
#       m_mon                m_monitor               -     @575
#         m_port             uvm_analysis_port       -     @600
#       m_seqr               m_sequencer             -     @424
#         rsp_export         uvm_analysis_export     -     @432
#         seq_item_export    uvm_seq_item_pull_imp   -     @538
#         arbitration_queue  array                   0     -
#         lock_queue         array                   0     -
#         num_last_reqs      integral                32    'd1
#         num_last_rsps      integral                32    'd1
#     m_agt_passive          m_passive_agent         -     @404
#       m_mon                m_monitor               -     @614
#         m_obs_port         uvm_analysis_port       -     @639
#       m_port               uvm_analysis_port       -     @624
#     m_sb                   scoreboard              -     @414
#       sb_exp_imp           uvm_analysis_imp        -     @648
#       sb_obs_imp           uvm_analysis_imp_obs    -     @657
# ------------------------------------------------------------
```

Active/Passive agents differ in get_type_name()

Split Agent Insts into Different Types



Modify get_type_name()
By adding layer to inheritance?