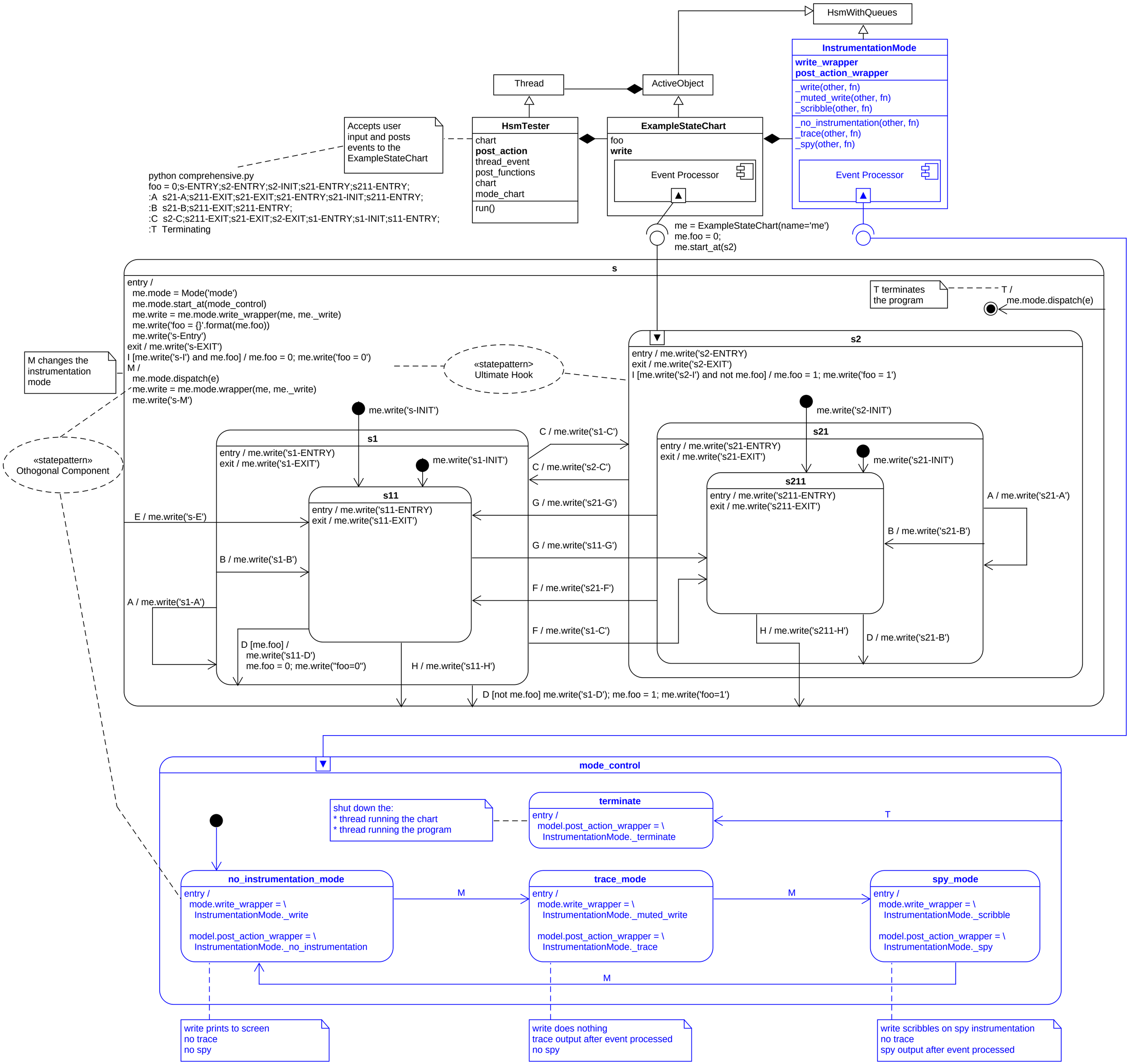
HsmWithQueues
InstrumentationMode
write_wrapper
post_action_wrapper
_write(other, fn)
_muted_write(other, fn)
_scribble(other, fn)
_no_instrumentation(other, fn)
_trace(other, fn)
_spy(other, fn)
Event Processor
Thread
ActiveObject
HsmTester
chart
post_action
thread_event
post_functions
chart
mode_chart
run()
ExampleStateChart
foo
write
Event Processor
Accepts user input and posts events to the ExampleStateChart
python comprehensive.py
foo = 0;s-ENTRY;s2-ENTRY;s2-INIT;s21-ENTRY;s211-ENTRY;
:A s21-A;s211-EXIT;s21-EXIT;s21-ENTRY;s21-INIT;s211-ENTRY;
:B s21-B;s211-EXIT;s211-ENTRY;
:C s2-C;s211-EXIT;s21-EXIT;s2-EXIT;s1-ENTRY;s1-INIT;s11-ENTRY;
:T Terminating
me = ExampleStateChart(name='me')
me.foo = 0;
me.start_at(s2)
s
entry /
me.mode = Mode('mode')
me.mode.start_at(mode_control)
me.write = me.mode.write_wrapper(me, me._write)
me.write('foo = {}'.format(me.foo))
me.write('s-Entry')
exit / me.write('s-EXIT')
I [me.write('s-I') and me.foo] / me.foo = 0; me.write('foo = 0')
M /
me.mode.dispatch(e)
me.write = me.mode.wrapper(me, me._write)
me.write('s-M')
M changes the instrumentation mode
«statepattern» Ultimate Hook
T terminates the program
T /
me.mode.dispatch(e)
s2
entry / me.write('s2-ENTRY)
exit / me.write('s2-EXIT')
I [me.write('s2-I') and not me.foo] / me.foo = 1; me.write('foo = 1')
me.write('s-INIT')
me.write('s2-INIT')
«statepattern» Othogonal Component
s1
entry / me.write('s1-ENTRY)
exit / me.write('s1-EXIT')
me.write('s1-INIT')
C / me.write('s1-C')
C / me.write('s2-C')
s21
entry / me.write('s21-ENTRY)
exit / me.write('s21-EXIT')
me.write('s21-INIT')
s11
entry / me.write('s11-ENTRY)
exit / me.write('s11-EXIT')
s211
entry / me.write('s211-ENTRY)
exit / me.write('s211-EXIT')
G / me.write('s21-G')
G / me.write('s11-G')
F / me.write('s21-F')
F / me.write('s1-C')
E / me.write('s-E')
B / me.write('s1-B')
A / me.write('s1-A')
A / me.write('s21-A')
B / me.write('s21-B')
H / me.write('s211-H')
D / me.write('s21-B')
D [me.foo] /
me.write('s11-D')
me.foo = 0; me.write("foo=0")
H / me.write('s11-H')
D [not me.foo] me.write('s1-D'); me.foo = 1; me.write('foo=1')
mode_control
shut down the:
* thread running the chart
* thread running the program
terminate
entry /
model.post_action_wrapper = \
InstrumentationMode._terminate
T
no_instrumentation_mode
entry /
mode.write_wrapper = \
InstrumentationMode._write
model.post_action_wrapper = \
InstrumentationMode._no_instrumentation
M
trace_mode
entry /
mode.write_wrapper = \
InstrumentationMode._muted_write
model.post_action_wrapper = \
InstrumentationMode._trace
M
spy_mode
entry /
mode.write_wrapper = \
InstrumentationMode._scribble
model.post_action_wrapper = \
InstrumentationMode._spy
M
write prints to screen
no trace
no spy
write does nothing
trace output after event processed
no spy
write scribbles on spy instrumentation
no trace
spy output after event processed