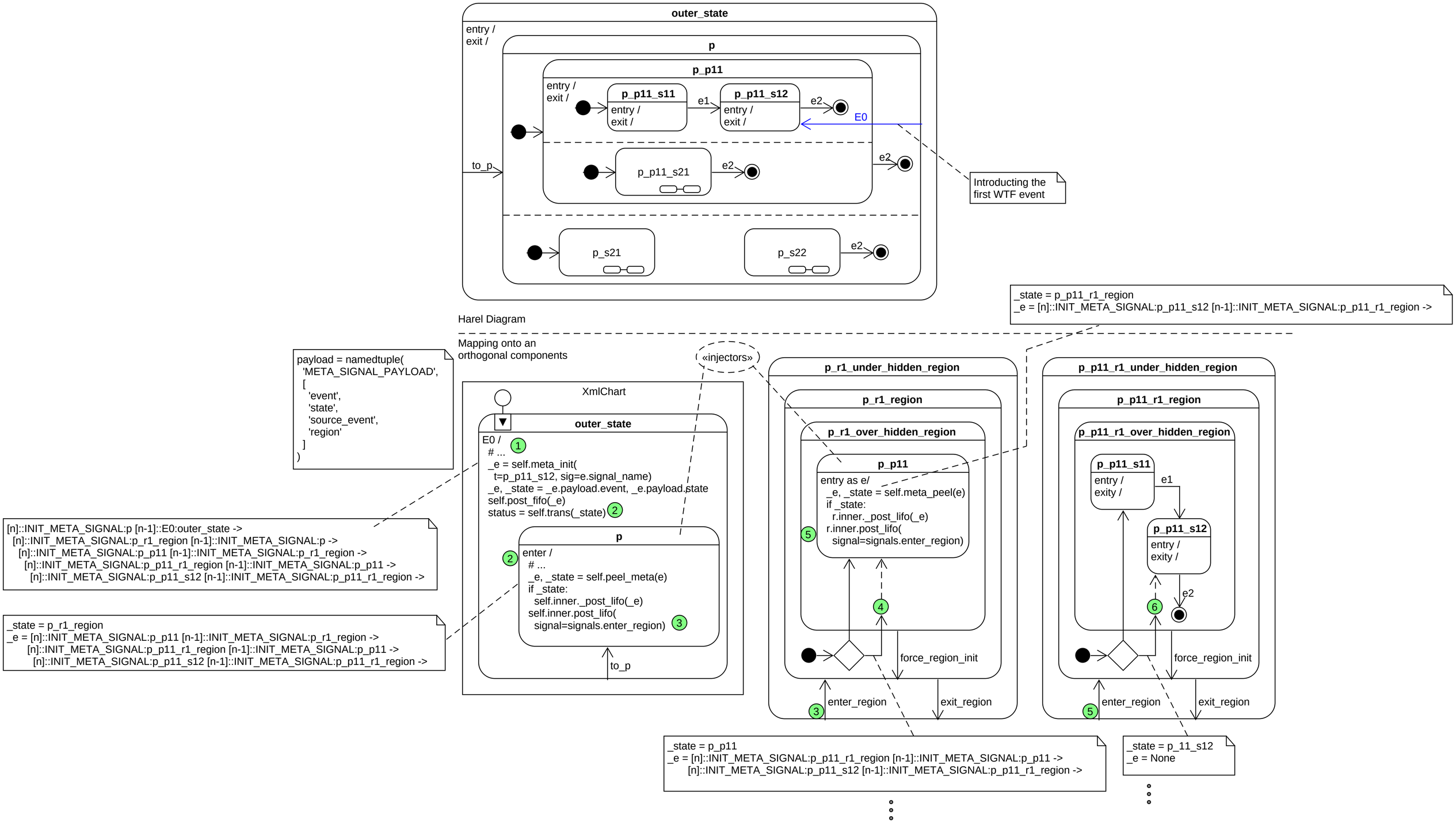
outer_state
entry /
exit /
p
p_p11
entry /
exit /
p_p11_s11
entry /
exit /
e1
p_p11_s12
entry /
exit /
e2
E0
to_p
p_p11_s21
e2
e2
p_s21
p_s22
e2
Introducting the
first WTF event
_state = p_p11_r1_region
_e = [n]::INIT_META_SIGNAL:p_p11_s12 [n-1]::INIT_META_SIGNAL:p_p11_r1_region ->
Harel Diagram
Mapping onto an
orthogonal components
«injectors»
payload = namedtuple(
'META_SIGNAL_PAYLOAD',
[
'event',
'state',
'source_event',
'region'
]
)
XmlChart
outer_state
E0 /
# ...
_e = self.meta_init(
t=p_p11_s12, sig=e.signal_name)
_e, _state = _e.payload.event, _e.payload.state
self.post_fifo(_e)
status = self.trans(_state)
p
enter /
...
_e, _state = self.peel_meta(e)
if _state:
self.inner._post_lifo(_e)
self.inner.post_lifo(
signal=signals.enter_region)
to_p
[n]::INIT_META_SIGNAL:p [n-1]::E0:outer_state ->
[n]::INIT_META_SIGNAL:p_r1_region [n-1]::INIT_META_SIGNAL:p ->
[n]::INIT_META_SIGNAL:p_p11 [n-1]::INIT_META_SIGNAL:p_r1_region ->
[n]::INIT_META_SIGNAL:p_p11_r1_region [n-1]::INIT_META_SIGNAL:p_p11 ->
[n]::INIT_META_SIGNAL:p_p11_s12 [n-1]::INIT_META_SIGNAL:p_p11_r1_region ->
_state = p_r1_region
_e = [n]::INIT_META_SIGNAL:p_p11 [n-1]::INIT_META_SIGNAL:p_r1_region ->
[n]::INIT_META_SIGNAL:p_p11_r1_region [n-1]::INIT_META_SIGNAL:p_p11 ->
[n]::INIT_META_SIGNAL:p_p11_s12 [n-1]::INIT_META_SIGNAL:p_p11_r1_region ->
p_r1_under_hidden_region
p_r1_region
p_r1_over_hidden_region
p_p11
entry as e/
_e, _state = self.meta_peel(e)
if _state:
r.inner._post_lifo(_e)
r.inner.post_lifo(
signal=signals.enter_region)
force_region_init
enter_region
exit_region
p_p11_r1_under_hidden_region
p_p11_r1_region
p_p11_r1_over_hidden_region
p_p11_s11
entry /
exity /
e1
p_p11_s12
entry /
exity /
e2
force_region_init
enter_region
exit_region
_state = p_p11
_e = [n]::INIT_META_SIGNAL:p_p11_r1_region [n-1]::INIT_META_SIGNAL:p_p11 ->
[n]::INIT_META_SIGNAL:p_p11_s12 [n-1]::INIT_META_SIGNAL:p_p11_r1_region ->
_state = p_11_s12
_e = None