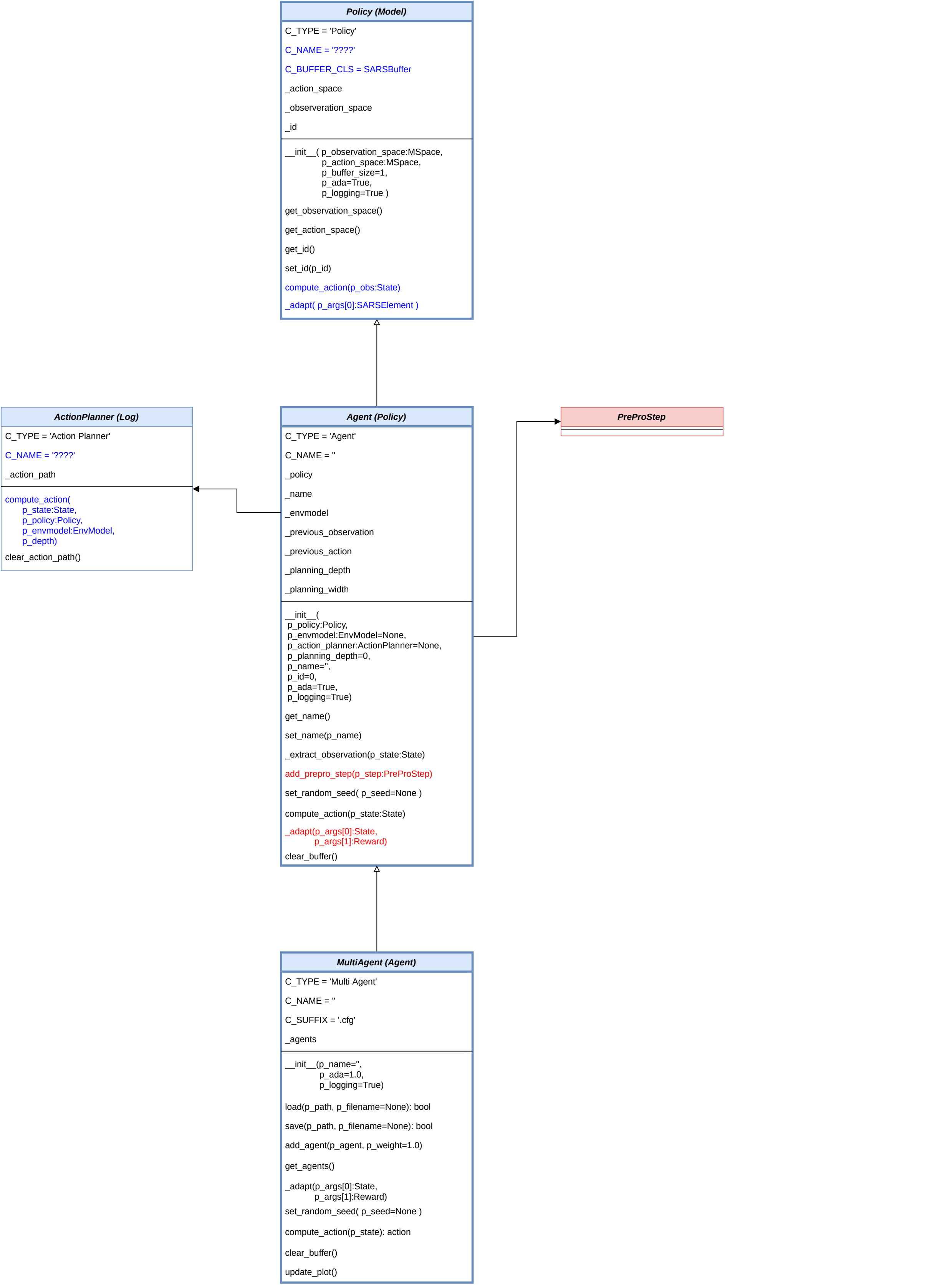
Policy (Model)
C_TYPE = 'Policy'
C_NAME = '????'
C_BUFFER_CLS = SARSBuffer
_action_space
_observeration_space
_id
__init__(p_observation_space:MSpace,
p_action_space:MSpace,
p_buffer_size=1,
p_ada=True,
p_logging=True)
get_observation_space()
get_action_space()
get_id()
set_id(p_id)
compute_action(p_obs:State)
_adapt(p_args[0]:SARSElement)
ActionPlanner (Log)
C_TYPE = 'Action Planner'
C_NAME = '????'
_action_path
compute_action(
p_state:State,
p_policy:Policy,
p_envmodel:EnvModel,
p_depth)
clear_action_path()
Agent (Policy)
C_TYPE = 'Agent'
C_NAME = ''
_policy
_name
_envmodel
_previous_observation
_previous_action
_planning_depth
_planning_width
__init__(
p_policy:Policy,
p_envmodel:EnvModel=None,
p_action_planner:ActionPlanner=None,
p_planning_depth=0,
p_name='',
p_id=0,
p_ada=True,
p_logging=True)
get_name()
set_name(p_name)
_extract_observation(p_state:State)
add_prepro_step(p_step:PreProStep)
set_random_seed(p_seed=None)
compute_action(p_state:State)
_adapt(p_args[0]:State,
p_args[1]:Reward)
clear_buffer()
PreProStep
MultiAgent (Agent)
C_TYPE = 'Multi Agent'
C_NAME = ''
C_SUFFIX = '.cfg'
_agents
__init__(p_name='',
p_ada=1.0,
p_logging=True)
load(p_path, p_filename=None): bool
save(p_path, p_filename=None): bool
add_agent(p_agent, p_weight=1.0)
get_agents()
_adapt(p_args[0]:State,
p_args[1]:Reward)
set_random_seed(p_seed=None)
compute_action(p_state): action
clear_buffer()
update_plot()
Legend:
Depends on
Inherits from
method()
New/inherited method/attribute to be implemented
method()
Method/attribute is already implemented