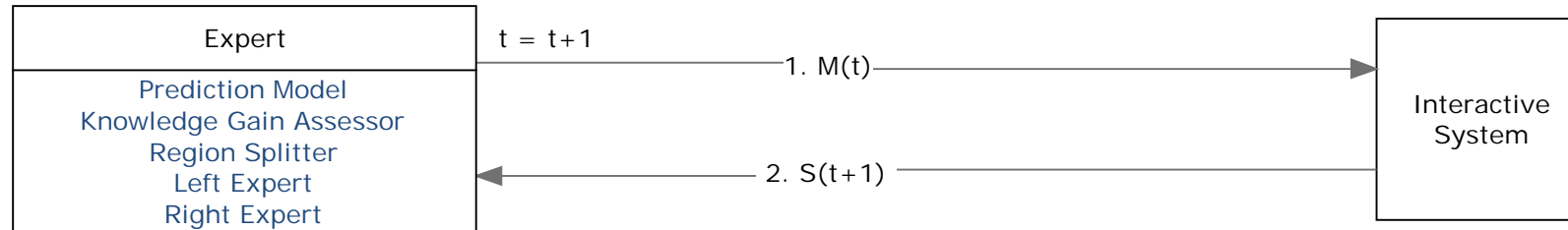# CBLA Block Diagram

Matthew T.K. Chan
2014-10-18

System Overview

Behaviour Pseudocode

```
t = 0
S(t) = S(0)
M(t) = M(0)

Loop:
    System → actuate(M(t))
    S(t+1) = System → report()

    S'(t+1) = Expert → Prediction_Model → predict(S(t), M(t))
    Expert → append(S(t), M(t), S(t+1), S'(t+1))
    Expert → split()
    M(t+1) = Expert → get_next_action(S(t+1))

    t = t+1
```

Expert → get_next_action() Pseudocode

M(t+1), Expected_Reward = get_new_action(S(t+1))

```
if is_leaf_expert():
    if S(t+1) is_possible():
        Expected_Reward = calc_expected_reward()
    else:
        Expected_Reward = -∞

    M(t+1) = mean({M*(0), M*(1), M*(2),…,M*(T)})

    return M(t+1), Expected_Reward

else:
    M_left, ER_left = Left_Expert → get_next_action(S(t+1))
    M_right, ER_right = Right_Expert → get_next_action(S(t+1))

    if is_exploring and ER_left > -∞ and ER_right > -∞:
        return random([M_left, ER_left] or [M_right, ER_right])
    else if ER_left > ER_right:
        return M_left, ER_left
    else:
        return M_right, ER_right
```
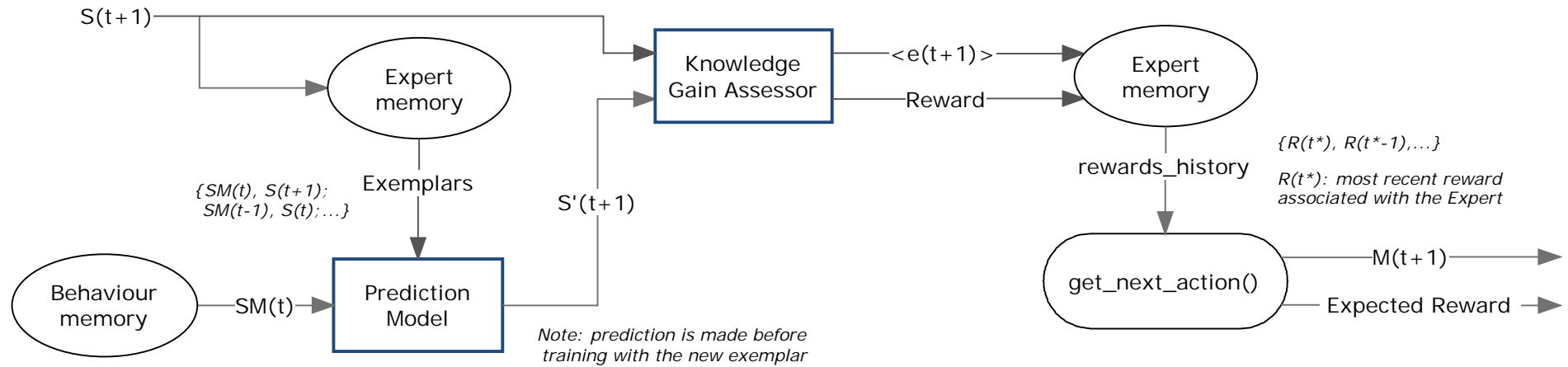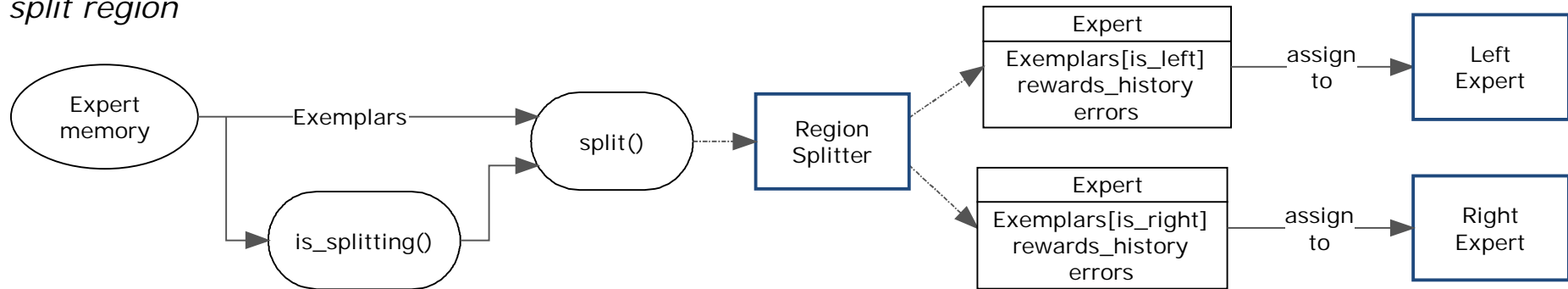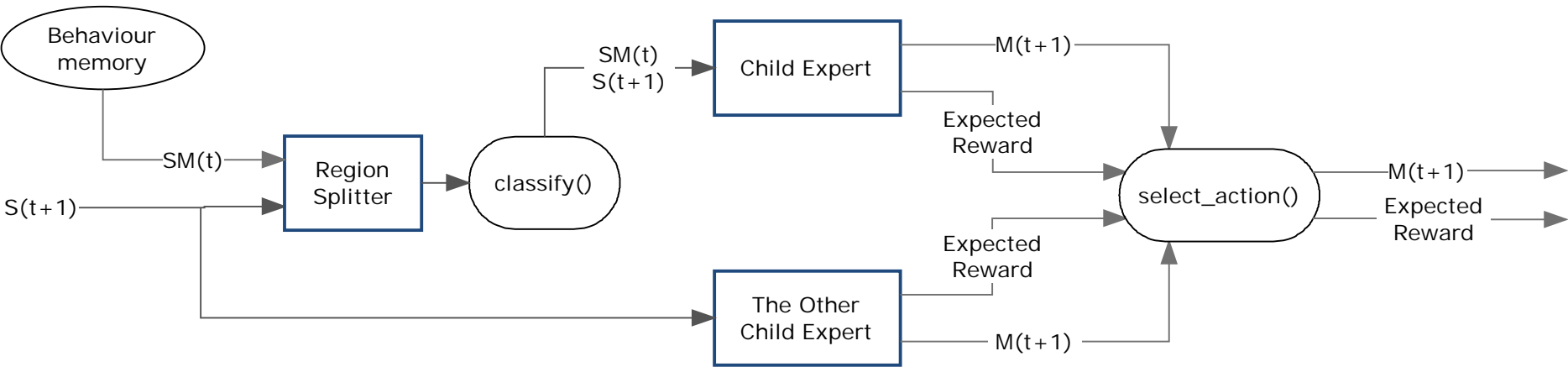
Leaf Expert

append exemplar

S(t+1)

Expert memory

Knowledge Gain Assessor

<e(t+1)>

Expert memory

Reward

{SM(t), S(t+1); SM(t-1), S(t);...}

Exemplars

S'(t+1)

rewards_history

{R(t), R(t*-1),...}*

R(t): most recent reward associated with the Expert*

Behaviour memory

SM(t)

Prediction Model

Note: prediction is made before training with the new exemplar

get_next_action()

M(t+1)

Expected Reward

split region

Expert memory

Exemplars

split()

Region Splitter

Expert
Exemplars[is_left] rewards_history errors

assign to

Left Expert

is_splitting()

Expert
Exemplars[is_right] rewards_history errors

assign to

Right Expert

Non-leaf Expert

append exemplar

Knowledge Gain Assessor