

Algorithms to Compute Appraisal Variables

Algorithm 1 (Relevance)

```
1: function ISEVENTRELEVANT(Event  $\varepsilon_t$ )  
  
2:   Initialize graph  $\mathcal{G}_t$  with current mental state  $\mathcal{S}_t$ .  
  
3:    $g_t \leftarrow \text{EXTRACTGOAL}(\mathcal{G}_t)$   
  
4:    $\mathcal{P}_t \leftarrow \text{EXTRACTPATHS}(\varepsilon_t, g_t)$   
  
5:   if ( $\mathcal{P}_t = \emptyset$ ) then  
6:     return FALSE  
7:   else  
8:      $\mathcal{U}_t \leftarrow \text{GETEVENTUTILITY}(\varepsilon_t, g_t)$   
9:     if ( $\mathcal{U}_t \geq \tau_e$ ) then  
10:      return TRUE  
11:    else  
12:      return FALSE  
13: end function
```

Algorithm 2 (Desirability)

```
1: function ISEVENTDESIRABLE(Event  $\varepsilon_t$ )

2:   Initialize graph  $\mathcal{G}_t$  with current mental state  $\mathcal{S}_t$ .

3:   if (topLevelTaskStatus() = ACHIEVED) then
4:     return TRUE
5:   else if (topLevelTaskStatus() = BLOCKED) then
6:     return FALSE
7:   else if (topLevelTaskStatus() = INPROGRESS) then

8:     if (currentTaskStatus() = ACHIEVED) then
9:       return TRUE
10:    else if (currentTaskStatus() = BLOCKED) then
11:      return FALSE
12:    else if (currentTaskStatus() = INPROGRESS) then
13:      return TRUE
14:    else if (currentTaskStatus() = UNKNOWN) then

15:      if (taskPreconditionStatus() = SATISFIED) then
16:        return TRUE
17:      else if (taskPreconditionStatus() = UNSATISFIED) then
18:        return FALSE
19:      else if (taskPreconditionStatus() = UNKNOWN) then

20:        if (doesContribute( $\varepsilon_t, \vec{g}_t$ ) = TRUE) then
21:          return TRUE
22:        else if (doesContribute( $\varepsilon_t, \vec{g}_t$ ) = FALSE) then

23:           $g_t \leftarrow \text{EXTRACTGOALS}(\mathcal{G}_t)$ 
24:           $\mathcal{U}_t \leftarrow \text{GETEVENTUTILITY}(\varepsilon_t, g_t)$ 
25:          if ( $\mathcal{U}_t \geq \tau_e$ ) then
26:            return TRUE
27:          else
28:            return FALSE
29:          end if

30:        end if
31:      end if
32:    end if
33:  end if
34: end function
```

Algorithm 3 (Expectedness)

```
1: function ISEVENTEXPECTED(Event  $\varepsilon_t$ )  
  
2:   Initialize graph  $\mathcal{G}_{t-1}$  with previous mental state  $\mathcal{S}_{t-1}$ .  
3:   Initialize graph  $\mathcal{G}_t$  with current mental state  $\mathcal{S}_t$ .  
  
4:    $g_{t-1} \leftarrow \text{EXTRACTGOALS}(\mathcal{G}_{t-1})$   
5:    $g_t \leftarrow \text{EXTRACTGOALS}(\mathcal{G}_t)$   
  
6:   if ( $g_t \neq g_{t-1}$ ) then  
7:     if (ISACHIEVED( $g_{t-1}$ ) = FALSE) then  
8:       return FALSE  
9:     else  
10:       $\mathcal{P}_t \leftarrow \text{EXTRACTPATHS}(\varepsilon_t, g_t)$   
11:      if ( $\mathcal{P}_t = \emptyset$ ) then  
12:        return FALSE  
13:      else  
14:         $\mathcal{U}_t \leftarrow \text{GETPATHUTILITY}(\mathcal{G}_t, g_t)$   
15:        if ( $\mathcal{U}_t \geq \tau_e$ ) then  
16:          return TRUE  
17:        else  
18:          return FALSE  
19:      else  
20:         $\mathcal{U}_t \leftarrow \text{GETPATHUTILITY}(\mathcal{G}_t, g_t)$   
21:         $\mathcal{U}_{t-1} \leftarrow \text{GETPATHUTILITY}(\mathcal{G}_t, g_{t-1})$   
22:        if ( $(\mathcal{U}_t - \mathcal{U}_{t-1}) \geq \tau_e$ ) then  
23:          return TRUE  
24:        else  
25:          return FALSE  
26: end function
```

Algorithm 4 (Controllability)

```
1: function ISEVENTCONTROLLABLE(Event  $\varepsilon_t$ )  
  
2:   Initialize graph  $\mathcal{G}_t$  with current mental state  $\mathcal{S}_t$ .  
  
3:    $g_t \leftarrow \text{EXTRACTGOALS}(\mathcal{G}_t)$   
4:    $\mathcal{P}_{g_t} \leftarrow \text{EXTRACTPATHS}(\varepsilon_t, g_t)$   
  
5:    $\alpha_{self/other}^{agency} \leftarrow \text{GETAGENCYVALUE}(\mathcal{P}_{g_t})$   
  
6:    $\beta_{self/other}^{autonomy} \leftarrow \text{GETAUTONOMYVALUE}(\mathcal{P}_{g_t})$   
  
7:    $\lambda_{succeeded/total}^{predecessors} \leftarrow \text{GETSUCCEEDEDPREDECESSORSRATIO}(\mathcal{P}_{g_t})$   
  
8:    $\mu_{available/required}^{inputs} \leftarrow \text{GETAVAILABLEINPUTRATIO}(\mathcal{P}_{g_t})$   
  
9:    $\mathcal{U}_t \leftarrow \frac{\omega_0 \cdot \alpha_{self/other}^{agency} + \omega_1 \cdot \beta_{self/other}^{autonomy} + \omega_2 \cdot \lambda_{succeeded/total}^{predecessors} + \omega_3 \cdot \mu_{available/required}^{inputs}}{\omega_0 + \omega_1 + \omega_2 + \omega_3}$   
  
10:  if ( $\mathcal{U}_t \geq \tau_e$ ) then  
11:    return TRUE  
12:  else  
13:    return FALSE  
14: end function
```

Algorithm 5 (Check Predecessors)

```
1: function GETSUCCEEDEDPREDECESSORSRATIO(Paths  $\mathcal{P}_{\vec{g}}^A$ )
2:    $count_{predecessor}^{succeeded} \leftarrow count_{predecessor}^{total} \leftarrow 0$ 
3:    $\Phi_{\vec{g}} \leftarrow \text{EXTRACTPREDECESSORS}(\mathcal{P}_{\vec{g}}^A)$ 
4:   for each  $\phi_{\vec{g}}^i \in \Phi_{\vec{g}}$  do
5:     if (ISUCCEEDED( $\phi_{\vec{g}}^i$ )) then
6:        $count_{predecessor}^{succeeded} \leftarrow count_{predecessor}^{succeeded} + 1$ 
7:        $count_{predecessor}^{total} \leftarrow count_{predecessor}^{total} + 1$ 
8:   return  $\langle count_{predecessor}^{succeeded}, count_{predecessor}^{total} \rangle$ 
9: end function
```

Algorithm 6 (Check Inputs)

```
1: function GETAVAILABLEINPUTRATIO(Paths  $\mathcal{P}_{\vec{g}}^A$ )
2:    $count_{input}^{available} \leftarrow count_{input}^{required} \leftarrow 0$ 
3:    $\mathcal{X}_{\vec{g}} \leftarrow \text{EXTRACTINPUTS}(\mathcal{P}_{\vec{g}}^A)$ 
4:   for each  $\chi_{\vec{g}}^i \in \mathcal{X}_{\vec{g}}$  do
5:     if (ISUCCEEDED( $\chi_{\vec{g}}^i$ )) then
6:        $count_{input}^{available} \leftarrow count_{input}^{available} + 1$ 
7:        $count_{input}^{required} \leftarrow count_{input}^{required} + 1$ 
8:   return  $\langle count_{input}^{available}, count_{input}^{required} \rangle$ 
9: end function
```

Algorithm 7 (Get Agency Value)

```
1: function GETAGENCYVALUE(Paths  $\mathcal{P}_{\bar{g}}^A$ )
2:    $count_{responsibility}^{self} \leftarrow count_{responsibility}^{other} \leftarrow 0$ 
3:    $\Theta_{\bar{g}} \leftarrow \text{EXTRACTPRECONDITIONS}(\mathcal{P}_{\bar{g}}^A)$ 
4:   for each  $\theta_{\bar{g}}^i \in \Theta_{\bar{g}}$  do
5:     if ( $\text{GETRESPONSIBLE}(\theta_{\bar{g}}^i) = \text{SELF}$ ) then
6:        $count_{responsibility}^{self} \leftarrow count_{responsibility}^{self} + 1$ 
7:     else
8:        $count_{responsibility}^{other} \leftarrow count_{responsibility}^{other} + 1$ 
9:   return  $\langle count_{responsibility}^{self}, count_{responsibility}^{other} \rangle$ 
10: end function
```

Algorithm 8 (Get Autonomy Value)

```
1: function GETAUTONOMYVALUE(Paths  $\mathcal{P}_{\bar{g}}^A$ )
2:    $\mathcal{A} \leftarrow \text{EXTRACTACTION}(\mathcal{P}_{\bar{g}}^A)$ 
3:    $\mathcal{R}_{\mathcal{A}} \leftarrow \text{GETRESPONSIBLE}(\mathcal{A})$ 
4:    $\mathcal{M}_{\mathcal{R}_{\mathcal{A}}} \leftarrow \text{GETMOTIVE}(\mathcal{R}_{\mathcal{A}})$ 
5:   if ( $\mathcal{M}_{\mathcal{R}_{\mathcal{A}}} \neq \emptyset$ ) then
6:     return MAX
7:   else
8:     return MIN
9: end function
```
