

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:    $\mathcal{P}_t \leftarrow \text{EXTRACTPATH}(\varepsilon_t, g_t)$   
  
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:           $g_t \leftarrow \text{EXTRACTGOAL}(\mathcal{G}_t)$ 
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:    $g_{t-1} \leftarrow \text{EXTRACTGOAL}(\mathcal{G}_{t-1})$   
5:    $g_t \leftarrow \text{EXTRACTGOAL}(\mathcal{G}_t)$   
  
6:   if ( $g_t \neq g_{t-1}$ ) then  
7:     if (ISACHIEVED( $g_{t-1}$ ) = FALSE) then  
8:       return FALSE  
9:     else  
10:       $\mathcal{P}_t \leftarrow \text{EXTRACTPATH}(\varepsilon_t, g_t)$   
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:         $\mathcal{U}_t \leftarrow \text{GETEVENTUTILITY}(\mathcal{G}_t, g_t)$   
21:        if ( $\mathcal{U}_t \geq \tau_e$ ) then  
22:          return TRUE  
23:        else  
24:          return FALSE  
25: 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:    $g_t \leftarrow \text{EXTRACTGOAL}(\mathcal{G}_t)$   
4:    $\mathcal{P}_{g_t} \leftarrow \text{EXTRACTPATH}(\varepsilon_t, g_t)$   
  
5:    $\alpha_{self/other}^{agency} \leftarrow \text{GETAGENCYRATIO}(g_t)$   
  
6:    $\beta_{self/other}^{autonomy} \leftarrow \text{GETAUTONOMYRATIO}(g_t)$   
  
7:    $\lambda_{succeeded/total}^{predecessors} \leftarrow \text{GETSUCCEEDEDPREDECESSORSRATIO}(g_t)$   
  
8:    $\mu_{available/required}^{inputs} \leftarrow \text{GETAVAILABLEINPUTRATIO}(g_t)$   
  
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 (Get Succeeded Predecessors Ratio)

```
1: function GETSUCCEEDEDPREDECESSORSRATIO(Goal  $g_t$ )

2:    $\Phi_g \leftarrow \text{EXTRACTPREDECESSORS}(g_t)$ 

3:   for each  $\phi_g^i \in \Phi_g$  do
4:     if (ISACHIEVED( $\phi_g^i$ )) then
5:        $count_{predecessor}^{achieved} \leftarrow count_{predecessor}^{achieved} + 1$ 

6:    $count_{predecessor}^{total} \leftarrow count_{predecessor}^{total} + 1$ 
7:   end for

8:   return  $\frac{count_{predecessor}^{achieved}}{count_{predecessor}^{total}}$ 

9: end function
```

Algorithm 6 (Get Available Input Ratio)

```
1: function GETAVAILABLEINPUTRATIO(Goal  $g_t$ )

2:    $\mathcal{X}_g \leftarrow \text{EXTRACTINPUTS}(g_t)$ 

3:   for each  $\chi_g^i \in \mathcal{X}_g$  do
4:     if (ISAVAILABLE( $\chi_g^i$ )) then
5:        $count_{input}^{available} \leftarrow count_{input}^{available} + 1$ 

6:    $count_{input}^{required} \leftarrow count_{input}^{required} + 1$ 
7:   end for

8:   return  $\frac{count_{input}^{available}}{count_{input}^{required}}$ 

9: end function
```

Algorithm 7 (Get Agency Value)

```
1: function GETAGENCYVALUE(Goal  $g_t$ )  
2:    $\Theta_g \leftarrow \text{EXTRACTCONTRIBUTINGTASKS}(g_t)$   
3:   for each  $\theta_g^i \in \Theta_g$  do  
4:     if (GETRESPONSIBLE( $\theta_g^i$ ) = SELF) then  
5:        $count_{responsibility}^{self} \leftarrow count_{responsibility}^{self} + 1$   
6:     else if (GETRESPONSIBLE( $\theta_g^i$ ) = BOTH) then  
7:        $count_{responsibility}^{self} \leftarrow count_{responsibility}^{self} + 0.5$   
8:     end for  
9:   return  $\frac{count_{responsibility}^{self}}{\Theta_g.size()}$   
10: end function
```

Algorithm 8 (Get Autonomy Value)

```
1: function GETAUTONOMYVALUE(Goal  $g_t$ )  
2:    $\mathcal{P}_{g_t} \leftarrow \text{EXTRACTPATH}(\varepsilon_t, g_t)$   
3:    $\mathcal{M}_{g_t} \leftarrow \text{EXTRACTMOTIVE}(\mathcal{P}_{g_t})$   
4:   if ( $\mathcal{M}_{g_t} \neq \emptyset$ ) then  
5:     if (GETMOTIVETYPE( $\mathcal{M}_{g_t}$ ) = INTERNAL) then  
6:       return 1.0  
7:     else if (GETMOTIVETYPE( $\mathcal{M}_{g_t}$ ) = EXTERNAL) then  
8:       return 0.5  
9:   else  
10:    return 0.0  
11: end function
```
