

## Algorithms to Compute Appraisal Variables

---

**Algorithm 1** (Relevance)

---

```
1: function ISEVENTRELEVANT(Events  $\varepsilon_t$ )  
  
2:   Initialize graph  $\mathcal{G}_t$  with current mental state  $\mathcal{S}_t$ .  
  
3:    $\vec{g}_t \leftarrow \text{EXTRACTGOAL}(\mathcal{G}_t)$   
  
4:    $\mathcal{P}_t \leftarrow \text{EXTRACTPATHS}(\varepsilon_t, \vec{g}_t)$   
  
5:   if ( $\mathcal{P}_t = \emptyset$ ) then  
6:     return 0  
7:   else  
8:      $\mathcal{U}_t \leftarrow \text{GETEVENTUTILITY}(\varepsilon_t, \vec{g}_t)$   
9:     if ( $\mathcal{U}_t \geq \tau_e$ ) then  
10:      return ( $\mathcal{U}_t$ )  
11:    else  
12:      return 0  
13: end function
```

---

---

**Algorithm 2** (Desirability)

---

```
1: function ISEVENTDESIRABLE(Events  $\varepsilon_t$ )  
  
2:   Initialize graph  $\mathcal{G}_t$  with current mental state  $\mathcal{S}_t$ .  
  
3:    $\vec{g}_t \leftarrow \text{EXTRACTGOALS}(\mathcal{G}_t)$   
  
4:   if (topLevelTaskStatus() = ACHIEVED) then  
5:     return 1.0  
6:   else if (topLevelTaskStatus() = BLOCKED) then  
7:     return -1.0  
8:   else if (topLevelTaskStatus() = INPROGRESS) then  
9:     if (currentTaskStatus() = ACHIEVED) then  
10:      return 0.75  
11:     else if (currentTaskStatus() = BLOCKED) then  
12:       return -0.75  
13:     else if (currentTaskStatus() = INPROGRESS) then  
14:       return 0.25  
15:     else if (currentTaskStatus() = UNKNOWN) then  
16:       if (taskPreconditionStatus() = SATISFIED) then  
17:         return 0.5  
18:       else if (taskPreconditionStatus() = UNSATISFIED) then  
19:         return -0.75  
20:       else if (taskPreconditionStatus() = UNKNOWN) then  
21:         if (doesContribute( $\varepsilon_t, \vec{g}_t$ ) = TRUE) then  
22:           return 0.0  
23:         else if (doesContribute( $\varepsilon_t, \vec{g}_t$ ) = FALSE) then  
24:           if (recipeApplicability( $\varepsilon_t, \vec{g}_t$ ) = APPLICABLE) then  
25:             return -0.5  
26:           else if (recipeApplicability( $\varepsilon_t, \vec{g}_t$ ) = INAPPLICABLE) then  
27:             return -0.75  
28:           else if (recipeApplicability( $\varepsilon_t, \vec{g}_t$ ) = UNKNOWN) then  
29:             return -0.25  
30: end function
```

---

---

**Algorithm 3** (Expectedness)

---

```
1: function ISEVENTEXPECTED(Events  $\varepsilon_t$ )

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( $g_{t-1}$ )) then
8:        $\mathcal{P}_t \leftarrow \text{EXTRACTPATHS}(\varepsilon_t, g_t)$ 
9:       if ( $\mathcal{P}_t = \emptyset$ ) then
10:        return FALSE
11:      else
12:         $\mathcal{U}_t \leftarrow \text{GETPATHUTILITY}(\mathcal{G}_t, g_t)$ 
13:         $\mathcal{U}_{t-1} \leftarrow \text{GETPATHUTILITY}(\mathcal{G}_t, g_{t-1})$ 
14:        if ( $\mathcal{U}_t - \mathcal{U}_{t-1} \geq \tau_e$ ) then
15:          return TRUE
16:        else
17:          return FALSE
18:      else ▷ Collaboration goal has achieved.
19:        return TRUE
20:    else ▷ Goals have not changed.
21:      return TRUE
22: end function
```

---

---

**Algorithm 4** (Controllability)

---

```
1: function ISEVENTCONTROLLABLE(Events  $\varepsilon_t$ )  
  
2:    $\alpha_{self/other}^{agency} \leftarrow \beta_{self/other}^{autonomy} \leftarrow 0$   
  
3:    $\lambda_{succeeded/failed}^{predecessors} \leftarrow \mu_{available/required}^{inputs} \leftarrow 0$   
  
4:   Initialize graph  $\mathcal{G}_t$  with current mental state  $\mathcal{S}_t$ .  
  
5:    $\vec{g}_t \leftarrow \text{EXTRACTGOALS}(\mathcal{G}_t)$   
6:    $\mathcal{P}_{\vec{g}_t} \leftarrow \text{EXTRACTPATHS}(\varepsilon_t, \vec{g}_t)$   
  
7:    $\alpha_{self/other}^{agency} \leftarrow \text{GETAGENCYVALUE}(\mathcal{P}_{\vec{g}_t})$   
  
8:    $\beta_{self/other}^{autonomy} \leftarrow \text{GETAUTONOMYVALUE}(\mathcal{P}_{\vec{g}_t})$   
  
9:    $\lambda_{succeeded/total}^{predecessors} \leftarrow \text{GETSUCCEEDEDPREDECESSORSRATIO}(\mathcal{P}_{\vec{g}_t})$   
  
10:   $\mu_{available/required}^{inputs} \leftarrow \text{GETAVAILABLEINPUTRATIO}(\mathcal{P}_{\vec{g}_t})$   
11:   $\mathcal{U}_t \leftarrow \frac{\omega_0 \cdot \alpha_{self/other}^{agency} + \omega_1 \cdot \beta_{self/other}^{autonomy} + \omega_2 \cdot \lambda_{succeeded/total}^{predecessors} + \omega_3 \cdot \mu_{available/required}^{inputs}}{\omega_0 + \omega_1 + \omega_2 + \omega_3}$   
  
12:  if ( $\mathcal{U}_t \geq \tau_e$ ) then  
13:    return TRUE  
14:  else  
15:    return FALSE  
16: 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
```

---