Algorithms to Compute Appraisal Variables

Algorithm 1 (Relevance)

```
function IsEventRelevant(Events $\varepsilon_t$)

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

    $\vec{g}_t \leftarrow$ extractGoal($\mathcal{G}_t$)

    $\mathcal{P}_t \leftarrow$ extractPaths($\varepsilon_t, \vec{g}_t$)

    if ($\mathcal{P}_t = \emptyset$) then
        return 0
    else
        $\mathcal{U}_t \leftarrow$ getEventUtility($\varepsilon_t, \vec{g}_t$)
        if ($\mathcal{U}_t \geq \tau_e$) then
            return ($\mathcal{U}_t$)
        else
            return 0
end function
```

Algorithm 2 (Desirability)

```
function IsEventDesirable($Events\ \varepsilon_t$)

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

    $\vec{g}_t \leftarrow$ extractGoals($\mathcal{G}_t$)

    if ($topLevelTaskStatus()$ = ACHIEVED) then
        return 1.0
    else if ($topLevelTaskStatus()$ = BLOCKED) then
        return -1.0
    else if ($topLevelTaskStatus()$ = INPROGRESS) then
        if ($currentTaskStatus()$ = ACHIEVED) then
            return 0.75
        else if ($currentTaskStatus()$ = BLOCKED) then
            return -0.75
        else if ($currentTaskStatus()$ = INPROGRESS) then
            return 0.25
        else if ($currentTaskStatus()$ = UNKNOWN) then
            if ($taskPreconditionStatus()$ = SATISFIED) then
                return 0.5
            else if ($taskPreconditionStatus()$ = UNSATISFIED) then
                return -0.75
            else if ($taskPreconditionStatus()$ = UNKNOWN) then
                if ($doesContribute(\varepsilon_t, \vec{g}_t)$ = TRUE) then
                    return -0.5
                else if ($doesContribute(\varepsilon_t, \vec{g}_t)$ = FALSE) then
                    if ($recipeApplicability(\varepsilon_t, \vec{g}_t)$ = APPLICABLE) then
                        return -0.5
                    else if ($recipeApplicability(\varepsilon_t, \vec{g}_t)$ = INAPPLICABLE) then
                        return -0.75
                    else if ($recipeApplicability(\varepsilon_t, \vec{g}_t)$ = UNKNOWN) then
                        return -0.25
end function
```

Algorithm 3 (Get Utterance Utility)

```
function GetUtteranceUtility(Paths $\mathcal{P}_{\vec{g}_t}$, Paths $\mathcal{P}_{\vec{g}_{t-1}}$)

    $\mathcal{U}_{\vec{g}_t} \leftarrow \sum_{\mathcal{P}^{U_i}_{\vec{g}_t} \in \mathcal{P}^{U}_{\vec{g}_t}}$ GetMentalStateUtility($\mathcal{P}^{U_i}_{\vec{g}_t}$)
    $\hat{\mathcal{U}}_{\vec{g}_t} \leftarrow \frac{\mathcal{U}_{\vec{g}_t}}{\left|\mathcal{P}^{U}_{\vec{g}_t}\right|}$
    $\mathcal{U}_{\vec{g}_{t-1}} \leftarrow \sum_{\mathcal{P}^{U_i}_{\vec{g}_{t-1}} \in \mathcal{P}^{U}_{\vec{g}_{t-1}}}$ GetMentalStateUtility($\mathcal{P}^{U_i}_{\vec{g}_{t-1}}$)
    $\hat{\mathcal{U}}_{\vec{g}_{t-1}} \leftarrow \frac{\mathcal{U}_{\vec{g}_{t-1}}}{\left|\mathcal{P}^{U}_{\vec{g}_{t-1}}\right|}$
    return ($\hat{\mathcal{U}}_{\vec{g}_t}$ - $\hat{\mathcal{U}}_{\vec{g}_{t-1}}$)
end function
```

Algorithm 4 (Get Action Utility)

```
function GetActionUtility(Paths $\mathcal{P}_{\vec{g}_t}$, Paths $\mathcal{P}_{\vec{g}_{t-1}}$)

    $\theta \leftarrow$ GetPreconditionUtility($\mathcal{P}_{\vec{g}_t}, \mathcal{P}_{\vec{g}_{t-1}}$)
    $\psi \leftarrow$ GetPostconditionUtility($\mathcal{P}_{\vec{g}_t}, \mathcal{P}_{\vec{g}_{t-1}}$)

    $\vec{\omega} \leftarrow$ GetUtilityWeights()

    $\Delta U \leftarrow \frac{\omega_0 \cdot \theta + \omega_1 \cdot \psi}{\omega_0 + \omega_1}$

    return $\Delta U$
end function
```

Algorithm 5 (Get Emotion Utility)

```
function GetEmotionUtility(Paths $\mathcal{P}_{\vec{g}_t}$, Paths $\mathcal{P}_{\vec{g}_{t-1}}$)

    $\mathcal{P}^{\mathcal{E}}_{\vec{g}_{t-1}} \leftarrow$ ExtractEmotionPaths($\mathcal{P}_{\vec{g}_{t-1}}$)
    $\mathcal{P}^{\mathcal{E}}_{\vec{g}_t} \leftarrow$ ExtractEmotionPaths($\mathcal{P}_{\vec{g}_t}$)

    $\mathcal{E}^{other}_{t-1} \leftarrow$ ExtractOthersEmotion($\mathcal{P}^{\mathcal{E}}_{\vec{g}_{t-1}}$)
    $\mathcal{E}^{other}_{t} \leftarrow$ ExtractOthersEmotion($\mathcal{P}^{\mathcal{E}}_{\vec{g}_t}$)

    $\Upsilon_{t-1} \leftarrow$ GetSomaticMarkers($\mathcal{E}^{other}_{t-1}$)
    $\Upsilon_{t} \leftarrow$ GetSomaticMarkers($\mathcal{E}^{other}_{t}$)

    $\Delta U \leftarrow \frac{\Bigg(\bigg(\Big(\big((\Upsilon^A_t - \Upsilon^A_{t-1}) \times 15\big) + (\Upsilon^V_t - \Upsilon^V_{t-1})\Big) \times 15\bigg) + (\Upsilon^S_t - \Upsilon^S_{t-1})\Bigg) \times 15 + (\Upsilon^I_t - \Upsilon^I_{t-1})}{15^4}$

    return $\Delta U$
end function
```

Algorithm 6 (Get Precondition Utility)

```
function GetPreconditionUtility($Paths\ \mathcal{P}_{\vec{g}_t}$, $Paths\ \mathcal{P}_{\vec{g}_{t-1}}$)
    $ratio^{sat}_{\vec{g}_t/\vec{g}_{t-1}} \leftarrow ratio^{sat/total}_{\vec{g}_t} \leftarrow 0$
    $\mathcal{P}^A_{\vec{g}_t} \leftarrow$ ExtractActionPaths($\mathcal{P}_{\vec{g}_t}$)
    if $(\vec{g}_t = \vec{g}_{t-1})$ then
        if $(\mathcal{P}^A_{t-1} \neq \emptyset)$ OR $(\mathcal{P}^A_t \neq \emptyset)$ then
            $\mathcal{P}^A_{\vec{g}_{t-1}} \leftarrow$ ExtractActionPaths($\mathcal{P}_{\vec{g}_{t-1}}$)
            $Precond^{sat}_{\mathcal{P}^A_{\vec{g}_t}} \leftarrow$ CheckPreconditions($\mathcal{P}^A_{\vec{g}_t}$)
            $Precond^{sat}_{\mathcal{P}^A_{\vec{g}_{t-1}}} \leftarrow$ CheckPreconditions($\mathcal{P}^A_{\vec{g}_{t-1}}$)
            $ratio^{sat}_{\vec{g}_t/\vec{g}_{t-1}} \leftarrow \omega_0 \cdot \frac{Precond^{sat}_{\mathcal{P}^A_{\vec{g}_t}}}{Precond^{sat}_{\mathcal{P}^A_{\vec{g}_{t-1}}}|_{\neq 0}} + \omega_1 \cdot \frac{Precond^{sat}_{\mathcal{P}^A_{\vec{g}_t}}}{Precond^{total}_{\mathcal{P}^A_{\vec{g}_t}}|_{\neq 0}}$
            return $ratio^{sat}_{\vec{g}_t/\vec{g}_{t-1}}$
        else
            return 0
    else
        if $(\mathcal{P}^A_t \neq \emptyset)$ then
            $Precond_{\mathcal{P}^A_{\vec{g}_t}} \leftarrow$ CheckPreconditions($\mathcal{P}^A_{\vec{g}_t}$)
            $ratio^{sat/total}_{\vec{g}_t} \leftarrow \frac{Precond^{sat}_{\mathcal{P}^A_{\vec{g}_t}}}{Precond^{total}_{\mathcal{P}^A_{\vec{g}_t}}|_{\neq 0}}$
            return $ratio^{sat/total}_{\vec{g}_t}$
        else
            return 0
end function
```

Algorithm 7 (Check Preconditions)

```
function CheckPreconditions($Paths\ \mathcal{P}^A_{\vec{g}}$)

    $count^{sat}_{precond} \leftarrow count^{total}_{precond} \leftarrow 0$

    $\Theta_{\vec{g}} \leftarrow$ ExtractPreconditions($\mathcal{P}^A_{\vec{g}}$)

    for each $\theta^i_{\vec{g}} \in \Theta_{\vec{g}}$ do
        if (IsSatisfied($\theta^i_{\vec{g}}$)) then
            $count^{sat}_{precond} \leftarrow count^{sat}_{precond} + 1$
        $count^{total}_{precond} \leftarrow count^{total}_{precond} + 1$

    return $\langle count^{sat}_{precond}, count^{total}_{precond} \rangle$
end function
```

Algorithm 8 (Get Postcondition Utility)

```
function GetPostconditionUtility($Paths\ \mathcal{P}_{\vec{g}_t}$, $Paths\ \mathcal{P}_{\vec{g}_{t-1}}$)

    $\mathcal{P}^A_{\vec{g}_t} \leftarrow$ ExtractActionPaths($\mathcal{P}_{\vec{g}_t}$)

    if ($\vec{g}_t = \vec{g}_{t-1}$) then
        if ($\mathcal{P}^A_{t-1} \neq \emptyset$) OR ($\mathcal{P}^A_t \neq \emptyset$) then

            $\mathcal{P}^A_{\vec{g}_{t-1}} \leftarrow$ ExtractActionPaths($\mathcal{P}_{\vec{g}_{t-1}}$)

            $\Psi_{\vec{g}_t} \leftarrow$ ExtractPostconditions($\mathcal{P}^A_{\vec{g}_t}$)
            $\Psi_{\vec{g}_{t-1}} \leftarrow$ ExtractPostconditions($\mathcal{P}^A_{\vec{g}_{t-1}}$)

            if (IsSatisfied($\Psi_{\vec{g}_t}$)) then
                if (IsSatisfied($\Psi_{\vec{g}_{t-1}}$)) then
                    return $\gamma^t$                    ▷ Discount rate: $0 \leq \gamma \leq 1$
                else
                    return 1
            else
                if IsSatisfied($\Psi_{\vec{g}_{t-1}}$)) then
                    return $-1$
                else
                    return 0
end function
```

Algorithm 9 (Expectedness)

```
function IsEventExpected(MentalStates $\mathcal{S}_t$, Event $\varepsilon_t$)

    Initialize graph $\mathcal{G}_{t-1}$ with previous mental state $\mathcal{S}_{t-1}$.
    Initialize graph $\mathcal{G}_t$ with current mental state $\mathcal{S}_t$.

    $\vec{g}_{t-1} \leftarrow$ extractGoals($\mathcal{G}_{t-1}$)
    $\vec{g}_t \leftarrow$ extractGoals($\mathcal{G}_t$)

    $\mathcal{P}_{t-1} \leftarrow$ extractPaths($\mathcal{G}_{t-1}, \vec{g}_{t-1}$)
    $\mathcal{P}_t \leftarrow$ extractPaths($\mathcal{G}_t, \vec{g}_t$)

    if ($\vec{g}_t \neq \vec{g}_{t-1}$) then
        if (IsAchieved($g^c_{t-1}$)) then
            if ($\neg$IsAchieved($g^p_{t-1}$)) then
                $\mathcal{P}'_{t-1} \leftarrow$ extractPaths($\mathcal{G}_{t-1}, \vec{g}_t$)
                if ($\mathcal{P}'_{t-1} = \emptyset$) then
                    return FALSE
                else
                    $\mathcal{U}'_{t-1} \leftarrow$ getUtilities($\mathcal{G}_{t-1}, \vec{g}_t$)
                    $\mathcal{U}_{t-1} \leftarrow$ getUtilities($\mathcal{G}_{t-1}, \vec{g}_{t-1}$)
                    if ($\mathcal{U}'_{t-1} - \mathcal{U}_{t-1}$) $\geq \tau_e$ then
                        return TRUE
                    else
                        return FALSE
            else                ▷ Collaborative and personal goals have achieved.
                return TRUE
        else                                ▷ Collaborative goal has not achieved.
            return FALSE
    else                                          ▷ Goals have not changed.
        return TRUE
end function
```

Algorithm 10 (Likelihood)

```
function GetEventsLikelihood($MentalStates\ \mathcal{S}_t$, $Event\ \varepsilon_t$)

    $\mathcal{G}_t \leftarrow$ updateGraph($\mathcal{S}_t, \varepsilon_t$)
    $\vec{g}_t \leftarrow$ extractGoals($\mathcal{G}_t$)
    $\mathcal{P}_t \leftarrow$ extractPaths($\mathcal{G}_t, \vec{g}_t$)

    if ($\mathcal{P}_t = \emptyset$) then
        return 0
    else
        for each $\mathcal{P}_t^i \in \mathcal{P}_t$ do
            for each $\varepsilon_{t+1}^j \in \varepsilon_{t+1}$ do
                $\vec{P}\left(\varepsilon_{t+1}^j \middle| \mathcal{F}(\mathcal{P}_t^i)\right) = \frac{\vec{P}\left(\mathcal{F}(\mathcal{P}_t^i) \middle| \varepsilon_{t+1}^j\right)\vec{P}(\varepsilon_{t+1}^j)}{\sum_{e_{t+1}^k \in \varepsilon_{t+1}^j} \vec{P}\left(\mathcal{F}(\mathcal{P}_t^i) \middle| e_{t+1}^k\right)}$

        return $\vec{P}\left(\varepsilon_{t+1} \middle| \mathcal{F}(\mathcal{P}_t^i)\right)$
end function
```

Algorithm 11 (Controllability)

```
function $\textsc{IsEventControllable}(\textit{MentalStates } \mathcal{S}_t, \textit{ Event } \varepsilon_t)$
    $\alpha^{agency}_{self/other} \leftarrow \beta^{autonomy}_{self/other} \leftarrow 0$
    $\kappa^{preconditions}_{satisfied/total} \leftarrow \lambda^{predecessors}_{succeeded/failed} \leftarrow \mu^{inputs}_{available/required} \leftarrow 0$
    Initialize graph $\mathcal{G}_t$ with current mental state $\mathcal{S}_t$.
    $\vec{g}_t \leftarrow \textsc{extractGoals}(\mathcal{G}_t)$
    $\mathcal{P}_{\vec{g}_t} \leftarrow \textsc{extractPaths}(\mathcal{G}_t, \vec{g}_t)$
    $\mathcal{P}^{A}_{\vec{g}_t} \leftarrow \textsc{ExtractActionPaths}(\mathcal{P}_{\vec{g}_t})$
    $\varphi^{A}_{\vec{g}_t} \leftarrow \textsc{GetRecipeApplicability}(\mathcal{P}^{A}_{\vec{g}_t})$
    $\mathcal{P}^{\mathcal{E}}_{\vec{g}_t} \leftarrow \textsc{ExtractEmotionPaths}(\mathcal{P}_{\vec{g}_t})$
    $\mathcal{E}^{other}_t \leftarrow \textsc{ExtractOthersEmotion}(\mathcal{P}^{\mathcal{E}}_{\vec{g}_t})$
    $\mathcal{I}^{emotion}_{other} \leftarrow \textsc{GetEmotionIntensity}(\mathcal{E}^{other}_t)$
    $\Upsilon_t \leftarrow \textsc{GetSomaticMarkers}(\mathcal{E}^{other}_t)$
    $\mathcal{V}^{\mathcal{E}}_t \leftarrow \textsc{GetEmotionValue}(\Upsilon_t, \varepsilon_t)$
    $Agency_{\vec{g}_t} \leftarrow \textsc{GetAgencyValue}(\mathcal{P}_{g_t})$
    $\alpha^{agency}_{self/other} \leftarrow {}^{Agency^{self}_{\vec{g}_t}}/_{Agency^{other}_{\vec{g}_t}}$
    $\beta^{autonomy}_{self/other} \leftarrow {}^{\textsc{GetAutonomyValue}(\mathcal{P}_{g^{self}_t})}/_{\textsc{GetAutonomyValue}(\mathcal{P}_{g^{other}_t})}$
    $Precond_{\mathcal{P}^{A}_{\vec{g}_t}} \leftarrow \textsc{CheckPreconditions}(\mathcal{P}^{A}_{\vec{g}_t})$
    $\kappa^{preconditions}_{satisfied/total} \leftarrow {}^{Precond^{satisfied}_{\mathcal{P}^{A}_{\vec{g}_t}}}/_{Precond^{total}_{\mathcal{P}^{A}_{\vec{g}_t}}}$
    $Predecessors_{\mathcal{P}^{A}_{\vec{g}_t}} \leftarrow \textsc{CheckPredecessors}(\mathcal{P}^{A}_{\vec{g}_t})$
    $\lambda^{predecessors}_{succeeded/total} \leftarrow {}^{Predecessors^{succeeded}_{\mathcal{P}^{A}_{\vec{g}_t}}}/_{Predecessors^{total}_{\mathcal{P}^{A}_{\vec{g}_t}}}$
    $Inputs_{\mathcal{P}^{A}_{\vec{g}_t}} \leftarrow \textsc{CheckInputs}(\mathcal{P}^{A}_{\vec{g}_t})$
    $\mu^{inputs}_{available/required} \leftarrow {}^{Inputs^{available}_{\mathcal{P}^{A}_{\vec{g}_t}}}/_{Inputs^{required}_{\mathcal{P}^{A}_{\vec{g}_t}}}$
    $\mathcal{U}_t \leftarrow \frac{\omega_0 \cdot \alpha^{agency}_{self/other} + \omega_1 \cdot \beta^{autonomy}_{self/other}}{\omega_0 \cdot + \omega_1 \cdot}$
        $+ \frac{\omega_2 \cdot \kappa^{preconditions}_{satisfied/total} + \omega_3 \cdot \lambda^{predecessors}_{succeeded/total} + \omega_4 \cdot \mu^{inputs}_{available/required}}{\omega_2 + \omega_3 + \omega_4}$
        $+ \frac{\omega_5 \cdot \mathcal{I}^{emotion}_{other} + \omega_6 \cdot \varphi^{A}_{\vec{g}_t} + \omega_7 \cdot \mathcal{V}^{\mathcal{E}}_t}{\omega_5 + \omega_6 + \omega_7}$

    **if** $(\mathcal{U}_t \geq \tau_e)$ **then**
        **return** TRUE
    **else**
        **return** FALSE
**end function**
```

Algorithm 12 (Check Predecessors)

```
**function** CheckPredecessors($Paths\ \mathcal{P}^{A}_{\vec{g}}$)

    $count^{succeeded}_{predecessor} \leftarrow count^{total}_{predecessor} \leftarrow 0$

    $\Phi_{\vec{g}} \leftarrow$ extractPredecessors($\mathcal{P}^{A}_{\vec{g}}$)

    **for each** $\phi^{i}_{\vec{g}} \in \Phi_{\vec{g}}$ **do**
        **if** (IsSucceeded($\phi^{i}_{\vec{g}}$)) **then**
            $count^{succeeded}_{predecessor} \leftarrow count^{succeeded}_{predecessor} + 1$
        $count^{total}_{predecessor} \leftarrow count^{total}_{predecessor} + 1$

    **return** $\langle count^{succeeded}_{predecessor}, count^{total}_{predecessor} \rangle$
**end function**
```

Algorithm 13 (Check Inputs)

```
**function** CheckInputs($Paths\ \mathcal{P}^{A}_{\vec{g}}$)

    $count^{available}_{input} \leftarrow count^{required}_{input} \leftarrow 0$

    $\mathcal{X}_{\vec{g}} \leftarrow$ extractInputs($\mathcal{P}^{A}_{\vec{g}}$)

    **for each** $\chi^{i}_{\vec{g}} \in \mathcal{X}_{\vec{g}}$ **do**
        **if** (IsSucceeded($\chi^{i}_{\vec{g}}$)) **then**
            $count^{available}_{input} \leftarrow count^{available}_{input} + 1$
        $count^{required}_{input} \leftarrow count^{required}_{input} + 1$

    **return** $\langle count^{available}_{input}, count^{required}_{input} \rangle$
**end function**
```

Algorithm 14 (Get Agency Value)

```
function GetAgencyValue($Paths\ \mathcal{P}_{\vec{g}}^{A}$)

    $count_{responsibility}^{self} \leftarrow count_{responsibility}^{other} \leftarrow 0$

    $\Theta_{\vec{g}} \leftarrow$ ExtractPreconditions($\mathcal{P}_{\vec{g}}^{A}$)

    for each $\theta_{\vec{g}}^{i} \in \Theta_{\vec{g}}$ do
        if (GetResponsible($\theta_{\vec{g}}^{i}$) = SELF) then
            $count_{responsibility}^{self} \leftarrow count_{responsibility}^{self} + 1$
        else
            $count_{responsibility}^{other} \leftarrow count_{responsibility}^{other} + 1$

    return $\langle count_{responsibility}^{self}, count_{responsibility}^{other} \rangle$
end function
```

Algorithm 15 (Get Autonomy Value)

```
function GetAutonomyValue($Paths\ \mathcal{P}_{\vec{g}}^{A}$)

    $\mathcal{A} \leftarrow$ ExtractAction($\mathcal{P}_{\vec{g}}^{A}$)
    $\mathcal{R}_{\mathcal{A}} \leftarrow$ GetResponsible($\mathcal{A}$)
    $\mathcal{M}_{\mathcal{R}_{\mathcal{A}}} \leftarrow$ GetMotive($\mathcal{R}_{\mathcal{A}}$)

    if ($\mathcal{M}_{\mathcal{R}_{\mathcal{A}}} \neq \emptyset$) then
        return MAX
    else
        return MIN
end function
```