

2017 年度卒業論文

エージェントの感情生起について
— 新たな感情の追加 —

奈良女子大学 生活環境学部 情報衣環境学科 生活情報通信科学コース 4 回生
新出研究室 14480207 浅井沙良

概要

近年、感情を付与したロボットの開発が進んでいる。ロボットが動的環境下において適切な感情を生起することによって、人間に近い行動選択をすることが望まれる。本研究では、人間らしい感情を付与した自律エージェントの実現を目指す。先行研究では、感情に関する理論 OCC theory の 22 種類の感情のうち Love と Hate 以外の 20 種類の感情を、程度の表現や時間経過に伴う変化も含めて実装している。それを基に、除外されていた Love と Hate の感情を実装に追加し、他者の望ましさについての感情の条件として利用することで感情の生起過程の改善を行った。また、条件として使用されていたふさわしさのオペレータ *Deserve* の定義を修正し、信念を伝えるアクションの機能の変更も行った。検証実験では、好き嫌いの感情を含めた複数の感情を生起するシナリオを用いてシミュレーションを行い、現実にも即した生起過程で感情が生起していることを確認した。最後に、より人間らしい感情を持つ自律的なエージェントの実現に向けて、課題点を述べる。

1 はじめに

近年、人間とロボットとのコミュニケーションが重要視されており、感情を付与したロボットの開発が進んでいる。ロボットに感情を付与することで、より人間らしい行動選択を自律的に行うことが期待されている。

従来ではロボットに感情を生起させる研究が広く行われている [1][2][3]。[1] では、動的環境下における目標達成の過程において、知性と感性を融合した感情生起を行うエージェントを提案している。しかし、自身に関連しない他エージェントの行動やその結果に対しては想定していないため、他者の状況を把握することで起こる感情（他者の良い結果、悪い結果に対する感情）や他者に対する感情（好き嫌いなど）は生起しない。また、[2][3] のように対話を行う際に感情を生起させるエージェントの研究は多く行われているが、それらはコミュニケーションをとる状況のみを想定した感情生起や、システムの感情を付加価値として使い心地を向上させることなどに重点を置いており、自身や他者の問題解決を行う状況における感情の生起は重要視していない。加えて、反応としての感情であり、一度生起した感情が後の感情生起には影響せず、生起要因を明示的に示すことが出来ないような場合が多い。今後ロボットと生活を共にしていくためには、ロボットが他者に対して多様な感情を持ち、行動やその結果、また以前に起きた感情を踏まえて、生起した理由を説明出来ることが必要になると考える。

周囲の環境が動的に変化する実世界において、問題解決のために行動するロボットの実現に BDI アーキテクチャが有効である [4]。BDI アーキテクチャとは、合理的行動のモデルである BDI モデル [5] による行為決定方式を計算機上で実現したものである。これにより、BDI エージェントの構築が可能であり、この構築には BDI logic と呼ばれる論理体系を用いる。

一方で、人間らしい感情の理論として、OCC theory [6] が挙げられる。OCC theory とは、人間の包括的な感情を 22 種類に分類した理論である。感情の生起条件が明確であるため、比較的理解しやすいとされる。また、OCC theory は BDI モデルが明示的に持つ信念や願望などの概念を用いて感情を特徴付けているため、両者は親和性が良く、どちらも記号論理で扱うことができる。実際、後述のように BDI アーキテクチャの既存の実装である Jason [7] 上の信念ベースを用いて実装が可能である。

OCC theory を BDI モデルに組み込むことで、Adam ら [8] は感情を論理式で形式化している。この研究では、OCC theory の感情のうち、形式化に一階述語論理を要する Love と Hate、つまり好き嫌いの感情のみが形式化されておらず、また、オペレータを限定しているため OCC theory の定義では感情の条件に関わるとされていても、形式化には反映されていない要素が幾つか存在している。

Adam らの形式化を基に、感情の生起条件を再定義し、OCC theory の感情生起を実装する研究がある

[9][10]。[9][10]の研究では、度合い付きの感情表現を持つエージェントの研究 [11] を基に、複数の感情の同時生起、時間による減衰を実装している。しかし、Adam らの形式化と同様に Love と Hate は実装されず、従って他者に対する感情を扱えないため、他者の望ましさについての感情の生起過程が現実には即したものはなかった。また、オペレータの *Deserve*(ふさわしさ) の定義と実装が不適切であり、アクションの機能や *Des*(望ましさ) の実装、度合いの設定に不具合があった。加えて、他者の感情推測を行う実装もされていない。

そこで、本研究では、人間らしいエージェントの実現に向けて、[9] [10] の研究を基に、Love と Hate の感情の追加、それらの感情を条件として利用した他者の望ましさについての感情の生起過程の改善、*Deserve*(ふさわしさ) の定義の検討、アクションの機能変更、他者の感情推測の追加、望ましさや度合いの設定についての不具合の修正を行った。実装は、BDI アーキテクチャを基礎とした Jason[7] で行った。なお、本研究は奈良女子大学生活環境学部情報衣環境学科生活情報通信科学コース 4 回生の塚本との共同研究であり、*Des*(望ましさ) と度合いの設定の不具合の修正、他者の感情推測については [12] で述べ、本論文では感情 Love と Hate の実装、それを利用した実装の改善、*Deserve*(ふさわしさ) の定義の検討とアクションの機能変更について述べる。

2 関連研究

本章では、感情の理論 OCC theory[6] と合理的行動のモデルである BDI モデル [5]、それらを用いて論理式を形式化した Adam らの研究 [8] について述べ、従来に実装された感情生起を行う自律エージェントの研究 [13][14][15][11][9][10] の概要と、実装の課題点を述べる。

2.1 OCC theory

本研究では、エージェントの感情生起に OCC theory[6] によって定義された感情を用いる。OCC theory とは、Ortony, Clore, Collins らが提唱した、心理学的見知を基に感情をモデル化した理論である。人間の包括的な感情を、信念や願望などの心的状態や、時間経過に基づいて分類している。感情の特徴付けが明確であり、論理モデルを用いた表現が可能であるため、計算機上での実装に適している。

OCC theory における 22 種類の感情のタイプは大きく 3 つのクラスに分類されており、その中の感情をさらにいくつかのグループに細分化している。以下にその分類を示す。

1. 結果の望ましさについて

(a) 自分にとっての望ましさについて

i. イベントに関して

— Joy(喜び), Distress(悲しみ)

ii. 予想に関して

A. 単なる予想に対して

— Hope(望み), Fear(恐れ)

B. 予想していたことが起こった

— Satisfaction(満足), FearConfirmed(恐れていたことが確定)

C. 予想していたことが起きなかった

— Relief(安堵), Disappointment(落胆)

(b) 他者にとっての望ましさについて

- i. 他者が良い結果を得た
 - HappyFor(ともに喜ぶ), Resentment(憤り)
 - ii. 他者が悪い結果を得た
 - SorryFor(ともに残念に思う), Gloating(ほくそ笑む)
2. 行動の賞賛について
- (a) 自分の行動に関して
 - Pride(誇り), Shame(羞恥心)
 - (b) 自分の望ましさに対する自分の行動に関して (1(a)i と 2(a) の混合型)
 - Gratification(満足), Remorse(後悔)
 - (c) 他者の行動に関して
 - Admiration(賞賛), Reproach(非難)
 - (d) 自分の望ましさに対する他者の行動に関して (1(a)i と 2(c) の混合型)
 - Gratitude(謝意), Anger(怒り)
3. 対象物に対する好き嫌いについて
- Love(好き), Hate(嫌い)

本研究で新たに実装に追加した感情は、「3. 対象物に対する好き嫌いについて」の感情である。また、それらの感情を利用して、「1.(b). 他者にとっての望ましさについて」の感情の生起過程を改善した。

2.2 BDI モデル

本研究は、BDI モデル [5] を基にして行った。BDI モデルとは、信念 (Belief)、願望 (Desire)、意図 (Intention) の 3 つの心的パラメータを用いて人間の合理的行動をモデル化したものである。達成すべき目標と手段を信念や願望によって選定するという意図の理論を行為決定に適用している。BDI モデルの行動様式により自律的に意思決定を行うエージェントを BDI エージェントと言い、BDI logic と呼ばれる様相論理体系を用いることで形式的に記述できる。また、BDI モデルの行為決定方式を計算機上で実現したものである BDI アーキテクチャの既存の実装に Jason[7] がある。本研究ではこの Jason を用いて実装を行う。

BDI モデルは自身の望ましさや時間経過についてなど、OCC theory による感情の特徴付けに用いられる概念を持っているため、両者は親和性が良い。また、どちらも記号論理で表現が可能であるため、BDI エージェントの実装手段をそのまま用いて OCC theory の感情を実装することが可能である。

2.3 Adam らによる形式化

Adam ら [8] は、BDI logic に複数の新たなオペレータを導入し、OCC theory の感情を BDI モデルに組み込むことで、感情を論理式で形式化している。新たに導入されたオペレータには、「不確実だが起こると期待されている事柄」などが挙げられる。Adam らは OCC theory の感情を 6 つのグループに分類することで、グループ毎に同じ形式の生起条件を用いた論理式を定義した。以下にそのグループ分けを示す。

- ① Well-being emotions : Joy, Distress
- ② Prospect emotions : Hope, Fear
- ③ Confirmation emotions : Satisfaction, FearConfirmed, Relief, Disappointment

- ④ Fortunes-of-others emotions : HappyFor, Resentment, SorryFor, Gloating
- ⑤ Attribution emotions : Pride, Shame, Admiration, Reproach,
Gratification, Remorse, Gratitude, Anger
- ⑥ Attraction emotions : Love, Hate

Adam らの形式化では命題論理を基にしている。一方で、⑥の感情は引数に対象物をとるため、形式化するには一階述語論理を用いる必要がある。そのため、Adam らは⑥の感情のみ論理式の形式化を行わなかった。

論理式の例として、①の *Joy*(喜び) と③の *Satisfaction*(満足)、④の *HappyFor*(ともに喜ぶ) の生起条件を以下に示す。

Joy(喜び) の生起条件は

$$Joy_i\varphi := Bel_i\varphi \wedge Des_i\varphi \quad (1)$$

である。これは「エージェント i がイベント φ が成立することを信じ、かつそのことがエージェント i にとって望ましいことであるならば、 i は φ の成立に対し、*Joy* という感情を生起する」ということを意味する。同じグループに属する *Distress*(悲しみ) の感情は、*Joy* の生起条件の $Des_i\varphi$ の部分が $Des_i\neg\varphi$ になり、イベント φ が望ましくない場合に生起する感情としている。

Satisfaction(満足) の生起条件は

$$Satisfaction_i\varphi := Bel_iPExpect_i\varphi \wedge Des_i\varphi \wedge Bel_i\varphi \quad (2)$$

である。これは *Joy* の生起条件に、「過去のある時点でイベント φ の成立を期待していた ($Bel_iPExpect_i\varphi$)」が条件に追加される。同様の論理式で、 φ が望ましくない場合には *FearConfermed*(恐れていたことが確定)、また、イベントの成立の部分について、 φ が起こらなかった場合には *Disappointment*(落胆)、 φ が望ましくなく、起こらなかった場合には *Relief*(安堵) が生起する。

HappyFor(ともに喜ぶ) の生起条件は

$$HappyFor_i\varphi := Bel_i\varphi \wedge Bel_iDes_j\varphi \wedge Des_iBel_j\varphi \quad (3)$$

である。これは「エージェント i がイベント φ が成立することを信じ、かつ、『 φ の成立がエージェント j にとって望ましい』ことであると i が信じ、かつ、『 j が φ の成立を信じる』ことが i にとって望ましいことであるならば、*HappyFor* という感情を生起する」ということを意味する。同様の論理式で、 φ が自身にとって望ましくない場合には *Resentment*(憤り)、他者にとって望ましくない場合には *Gloating*(ほくそ笑む)、自身と他者の両方にとって望ましくない場合には *SorryFor*(ともに残念に思う) が生起する。

2.4 感情を付与した自律エージェントの実装

Adam ら [8] が形式化した論理式を基にして、OCC theory の感情を付与した BDI エージェントの研究 [13][14][15][11][9][10] がある。これらの研究では、感情を生起させることにより、自律的に人間らしい行動選択を行うエージェントの実装を目指した。[13] では Adam らによる分類①②③ の感情を付与したエージェントを実装し、[14][15] によって未実装であった④⑤を実装した。[14][15] の研究では感情に度合いの表現がなかったため、[11] で感情に程度の強さを追加し、感情の連続生起も可能にした。その後の [9][10] では、複数の感情の同時生起、時間経過による減衰を実装した。

2.4.1 再定義された論理式について

先行研究 [9][10] では、Adam らの分類したグループ (⑥は含まない) ごとに論理式を再定義し、BDI アーキテクチャ既存の実装である Jason[7] の信念ベースを用いて感情生起を実装した。論理式を再定義するに当たって、Adam らの形式化した生起条件と生起する感情に度合いの表現を追加し、より OCC theory の定義に即した感情の生起条件になるよう新たなオペレータ *Effort*(努力)、*Deserve*(ふさわしさ) などを導入し、表記方法を変更して形式化を行っている。

新たに定義した論理式の例として *Satisfaction*(満足) と *HappyFor*(ともに喜ぶ) の生起条件を示す。

Satisfaction(満足) の生起条件は、

$$Satisfaction_{f(d,e,l)}^i \varphi := Bel^i PProb_i^i \varphi \wedge Bel^i PEffort_e^i \varphi \wedge Des_a^i \varphi \wedge Bel^i \varphi \wedge l \neq 1 \quad (4)$$

である。これは「エージェント i が過去のある時点でイベント φ が成立する可能性を l 程度信じ ($l \neq 1$)、かつ、 φ の成立のために e 程度努力し、かつ、 φ の成立がエージェント i にとって d 程度望ましいことであるならば、 i は φ の成立に対し、程度 $f(d,e,l)$ の *Satisfaction* という感情を生起する」ということを意味する。度合いを計算する関数 $f(d,e,l)$ は程度 d,e,l の増加関数とされているが、実装では程度 d,e,l を受け取り、加算して *Satisfaction* の程度として返している。

HappyFor(ともに喜ぶ) の生起条件は、

$$HappyFor_{f(di,dj)}^{i,j} \varphi := Bel^i \varphi \wedge Bel^i Des_{dj}^j \varphi \wedge Des_{di}^i Bel_j \varphi \wedge Bel^i Deserve^j \varphi \quad (5)$$

である。これは「エージェント i がイベント φ が成立することを信じ、かつ、『 φ の成立がエージェント j にとって dj 程度望ましい』ことであると i が信じ、かつ、『 j が φ の成立を信じる』ことが i にとって di 程度望ましく、かつ、 i にとって j に φ がふさわしいと感じるならば、程度 $f(di,dj)$ の *HappyFor* という感情を生起する」ということを意味する。度合いを計算する関数 $f(di,dj)$ は程度 di,dj の増加関数であり、実装では、程度 di,dj を受け取り、加算して *HappyFor* の程度として返している。このように、他の感情でも Adam らの論理式に程度の表現やオペレータを導入し、再定義を行った。

2.4.2 先行研究における実装内容

先行研究 [9][10] で実装されたシステムの場合図を図 1 に示し、動作について述べる。実装は Jason 上の信念ベースを用いて行い、図 1 では A,B,C の 3 エージェントの場合を示している。

システムの起動時には、初期信念内の信念をエージェントの信念ベースに追加する (a)。起動後、信念や行動選択が外部から追加されると「処理について」のプログラムで指定されたプランを実行する。

信念が追加された場合には「感情のチェック」のプログラムで検査し (b1)、追加された信念を信念ベースに保持させ (b0)、確認として出力する。感情の検査では、点線の矢印 (b2) が示している方を参照し、生起している感情があれば時刻と度合い付きで信念ベースに保持させ (b3)、現在生起しているすべての感情を度合い付きで出力する。

また、他エージェントに信念を伝えるアクション (!chat, !report) が行われる場合は、相手の信念ベースに、伝えた信念の追加が行われる (c)。例えば、*Des*(望ましさを伝えるアクション !chat を A が行った場合、友人の設定 friend(B) が A の信念ベースにあれば、 B の信念ベースに直前に A が感じた $Des^A \varphi$ が追加される。!report を行った場合は、同様の動作で *Bel*(イベント成立) が追加される。

外部入力で感情ごとの行動選択をするアクション (!act) が追加された場合は、「感情ごとの行動」のプログラムで指定された行動 (d1) を行う。この際、感情の検査を行うことで (d2,b2)、生起している感情を基に行動を選択する。例えば [9] の実装では、Adam らによる分類④に対するアクション !act を行った場合、感情 *Distress*(悲しみ) が生起していれば行動として「泣く」を出力する。

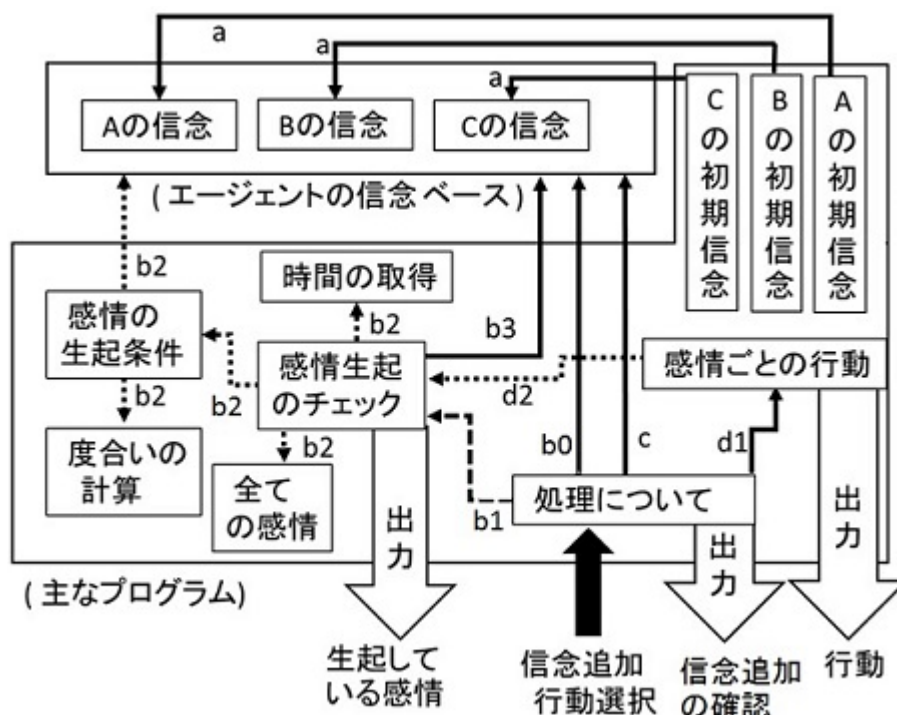


図1 感情生起システムの動作について

2.5 先行研究における課題

エージェントの感情生起を実装する研究 [9][10] では、Adam らが形式化した論理式を基にしているため、Love(好き) と Hate(嫌い) の感情は実装されていなかった。しかし、好き嫌いの感情は人間にとって重要な感情であり、意思決定に大きく影響すると考えられる。また、他の感情との関連も強い。従来は、シナリオに沿ったシミュレーションを行う際、他者への好き嫌いが描かれていないシナリオに限定していたため、検証範囲が限られていた。

④Fortunes-of-others emotions の感情 (他者の望ましさについての感情) では、他者に対する感情 Love と Hate を考慮せず、自身の望ましさを条件にしていたため感情が適切に生起しない問題があった。それに加え、他者の望ましさについての感情の条件として含まれていた *Deserve*(ふさわしさ) のオペレータが、実装ではふさわしさではなく *Des*(望ましさ) のオペレータと同じ意味として扱われており、アクションが行われた時の不具合を補う役割として追加されていた。しかし、OCC theory ではふさわしさは感情の度合いに関連するものとして定義されていたため、実装でも相手の望ましさではなく、ふさわしさとして扱われるべきだと考えられる。また、他エージェントに信念を伝えるアクションが、伝える相手や信念を指定することができず、2 エージェント以上でのやりとりを想定していなかった。(図1の例では A の信念に、friend(B), friend(C) の信念

がこの順に入っている場合には !chat を A が行っても、 B のエージェントにしか信念が伝えられない。))

そこで、本研究では新たに引数として対象物を取る Love と Hate の感情生起を実装し、それらを利用して他の感情の生起過程を改善することで、より人間らしい感情を生起するエージェントの実現を目指した。また、ふさわしさの定義の検討やアクションの機能の変更も行った。

3 感情 Love と Hate の実装

本研究では、従来の研究 [9][10] に加え、Love と Hate、つまり対象物に対する好き嫌いの感情を、他エージェントを引数に取る感情として実装した。本章ではその実装方法について述べる。

3.1 実装方針と論理式

Adam らの研究 [8] では、論理体系を命題論理に統一して形式化を行っていたため、先行研究 [9][10] でもそれに習い、命題論理で記述できる感情のみを扱っていた。しかし、実装で使用していた処理系 Jason[7] は述語論理を用いて記述できるため、実装が可能である。

OCC theory に基づく感情表現の論理モデルについての研究 [16] では、Love(好き) と Hate(嫌い) の感情を論理式で形式化している。そこで、[16] で定義された論理式を基に、感情生起の実装を行った。OCC theory[6] より、Love と Hate は、他者に対する魅力 (*Appealing*) と親密さ (*Familiar*) によって生起する感情だとされていた。[16] では、それらを新たなパラメータとして導入し、感情の生起条件としている。

[16] で定義された *Love* の生起条件は、

$$Love_{fL(a,f)}^i(j) := Appealing_a^i(j) \wedge Familiar_f^i(j) \quad (6)$$

である。これは「エージェント i がエージェント j のことを a 程度魅力的に感じており、かつ f 程度親密に思っている場合、 i は j に対して、 $fL(a,f)$ 程度、*Love* という感情を生起する」ということを意味する。度合いを計算する関数 $fL(a,f)$ は程度 a, f の増加関数としているが、程度 a, f の値を受け取り、加算して *Love* の程度として返す関数として実装した。

また、*Hate* の生起条件は

$$Hate_{fH(u,f)}^i(j) := Unappealing_u^i(j) \wedge Familiar_f^i(j) \quad (7)$$

である。これは、*Love* の条件の「 a 程度魅力的に感じている」部分が、「 u 程度魅力的ではない」という意味になる。関数 $fH(u,f)$ についても *Love* と同様に、程度 u, f を加算した値を程度として返す関数として実装した。

3.2 実装方法

前述の論理式の条件を実装に追加し、他エージェントを引数として *Love*(好き) と *Hate*(嫌い) が生起するよう実装した。*Love* と *Hate* は初期からエージェントが保持している感情だと考えられるため、エージェントの初期信念にオペレータ *Appealing*(魅力) または *Unappealing*(魅力的でない)、と *Familiar*(親密さ) の値を度合い付きで保持させることによって、システムの起動時から *Love* と *Hate* の感情を生起し、その後の保持を可能にした。それらの感情は時間による減衰や変化は起こさない感情として、不変的に保持させ、他の感情の生起条件として使用することを可能にした。また、相手エージェントに初期から魅力などを感じていない場合を想定し、その場合は、外部から新たに魅力や親密さの信念を追加することによって *Love* と *Hate* が生起する

生起した感情の出力結果 (起動時)

[tori] hate(tori,neko)1	← トリはネコのが 1 程度嫌い
[tori] love(tori,inu)0.30000000000000004	← トリはイヌのが 0.3 程度好き
[neko] hate(neko,inu)0.89999999999999999	← ネコはイヌのが 0.9 程度嫌い
[neko] hate(neko,tori)0.30000000000000004	← ネコはトリのが 0.3 程度嫌い
[inu] hate(inu,tori)0.30000000000000004	← イヌはトリのが 0.3 程度嫌い
[inu] love(inu,neko)0.9	← イヌはネコのが 0.9 程度好き

図 3 実行結果 (起動時)

4 他者に関する感情生起の改善

他者に対する感情 Love と Hate の保持が可能になったため、それらの感情を利用して④Fortunes-of-others emotions の感情 (他者の望ましさについての感情) について、OCC theory の定義を基に新たに論理式を定義し、生起過程の実装の改善を行った。

4.1 従来の定義の問題点と改善案

Adam らの形式化 [8] や先行研究 [9][10] では、④の感情の生起条件に自身の望ましさを利用していたが、それでは現実に即していないと考えられる。以下に④の *HappyFor*(ともに喜ぶ) の例を用いて、その問題点と改善案を示す。

HappyFor(ともに喜ぶ) は、OCC theory の特徴付け (2.1 章より) では「他者の望ましさについて」の「他者の良い結果に対する」感情とされていたが、Adam らの形式化では「自身がそのイベントの成立を望ましいと感じる」という条件が含まれていたため、OCC theory の定義に即していなかった。例えば、友人に関するイベントを「友人の宝くじが当たる」としたとき、それは自分とは直接関係のないイベントであるため、そのイベント自体を単体で望ましいと感じることはないはずである。望ましいと感じる場合は、その友人との関係 (親しいかなど) と、友人の望ましさ (宝くじがどれだけ当たってほしいか) を知っている前提があった場合であると想定される。しかし、[9] による実装では、自身の利益になるなどで自身にとっても望ましい場合 (友人の宝くじが当たれば、お金を分けてもらえるなど) や、外部入力者がシステムの外で友人との関係を考慮して望ましさを推測した場合に、望ましさが追加されていた。自分の利益が関係していることを理由に生起している場合は、「他者の良い結果に対する感情」として不適切である。また、他者への感情生起が可能になったため、他者に関するイベントへの望ましさの保持はシステムによって判断すべきことだと考えられる。ここで、他者への感情を利用した二通りの生起過程の改善案を挙げる。

一つめの案は、相手にとって良い結果か悪い結果かを判断することで相手の感情推測を行い、それに応じて相手との関係 (好き嫌い) によって感情を生起する方法である。例えば、友人が「宝くじが当たる」ことを望んでいると知っていれば、当たったとき友人が喜ぶため、その友人にある程度好意を抱いていたなら、ともに喜ぶはずである。もし、イベントが自身にとって望ましくないことで、好きな相手が良い結果を得たら少しはともに喜ぶはずであり、自身の望ましさを指標とする感情は本来別の分類とされているため、この生起条件が妥当だと考えられる。二つめの案は、相手の望ましいことであるならば、その相手との関係 (好き嫌い) によって自身も望ましいと感じるはずであり、その事象が成立したときに感情が生起する方法である。例えば、友人に好意を抱いている場合、友人が「宝くじが当たってほしい」なら自分もそうなって欲しいと感じる。そ

して、「宝くじが当たった」場合にはともに喜ぶはずである。この時の自身の望ましさは、好意が関連しているため通常の望ましさとは違う。このため、この望ましさを相手へのふさわしさとして定義することが出来ると考えた。これは、一つ目と同じ条件を用いているが、生起過程において重要視する部分が異なる。一つ目は他者の結果に対して予想される反応と他者との関係を考慮しているため、他者の感情を重視しており、二つ目は他者に起こるイベントの性質を他者との関係によって判断しているため、自身のイベントへの評価に重点を置いている。

一つめの案については、感情推測を利用して④の論理式を形式化し、実装した。(これを 4.2 章で述べる。) 二つめの案については、ふさわしさを定義した上で、④の論理式に組み込んで形式化した。(これを 4.3 章で述べる。) また、4.4 章では、一つめの案と二つめの案の定義した論理式について議論する。

4.2 Love,Hate と感情推測を用いた論理式の形式化

Love と Hate の感情と他者の感情推測を条件として論理式を再定義し、実装を改善した。感情推測については共同研究を行った塚本の論文 [12] で述べている。以下に、感情推測の論理式を示す。

[12] で定義された *Joy* の感情推測 *Guess_Joy* の生起条件は、

$$Guess_Joy_{f_{GJ}(dj)}^{i,j}\varphi := Bel^i Bel^j \varphi \wedge Bel^i Des_{dj}^j \varphi \quad (8)$$

である。これは従来の自身の感情生起の論理式を利用して、他者の感情推測を行う論理式である。*Guess_Distress* の場合も *Guess_Joy* の望ましさの部分を変えて、同じ方法で形式化している。実装では、自身の感情生起と同じ条件を用いて他者の感情推測を行い、自身の感情と同様に信念に追加している。これらの感情も *Love* と *Hate* と同様に、時間による減衰は起こさず保持させることで、他の感情の条件として利用できるよう実装した。度合いを計算する $f_{GJ}(dj)$ は、程度 dj の増加関数としているが、 dj の値を受け取り、そのまま *HappyFor* の程度として返す関数として実装した。

この感情推測の信念 *Guess_Joy*, *Guess_Distress* と、*Love*, *Hate* の感情を利用して④Fortunes-of-others emotions(他者の望ましさについての感情) の 4 つの感情の生起条件を、OCC theory を基にして再定義した。

新たに定義した *HappyFor*(ともに喜ぶ) の論理式は、

$$HappyFor_{f_H(f_L, f_{GJ})}^{i,j}\varphi := Love_{f_L}^i(j) \wedge Guess_Joy_{f_{GJ}}^{i,j}\varphi \quad (9)$$

である。これは「エージェント i がエージェント j のことを f_L 程度好きだと感じており、かつ、 i が j がイベント φ に対して f_{GJ} 程度喜んでいることを推測したならば、 $f_H(f_L, f_{GJ})$ 程度の *HappyFor* という感情を生起する」ということを意味する。度合いを計算する $f_H(f_L, f_{GJ})$ は、 f_L, f_{GJ} の増加関数としているが、 f_L, f_{GJ} の値を受け取り、加算して *HappyFor* の程度として返す関数として実装した。また、同じ分類に属する *Resentment*(憤り) は *Love* の部分が *Hate* である場合、*SorryFor*(ともに残念に思う) は *Guess_Joy* の部分が *Guess_Distress* である場合、*Gloating*(ほくそ笑む) は *Hate* と *Guess_Distress* を保持している場合に生起するという論理式で形式化した。

この *HappyFor* の論理式では、推測される他者の喜びの度合いと他者への好意の度合いによってともに喜ぶという生起過程を示している。2.4.1 章で例示した論理式 (5) ($HappyFor_{f(d_i, d_j)}^{i,j}\varphi := Bel^i \varphi \wedge Bel^i Des_{d_j}^j \varphi \wedge Des_{d_i}^i Bel_j \varphi \wedge Bel^i Deserve^j \varphi$) と表現は大きく異なるが、自身の望ましさとふさわしさの部分以外はほぼ同じ意味の条件になっている。ふさわしさについては再定義したため (4.3 章参照)、生起条件には含めていない。また、イベント成立の部分が異なっており、論理式 (5) では事象が成立したと自身が信じた ($Bel^i \varphi$) 時点で他

者が信じていなくても *HappyFor* が生起するという条件になっているが、本来は他者が成立を信じたことを自身が信じることで生起する感情だと考えられる。そのため、感情推測の条件とされている $Bel^i Bel^j \varphi$ の表現の方が妥当なため、論理式 (9) ではそのまま感情推測の論理式を条件として扱うことにした。

これにより、もとの意味を保持しながら、OCC theory で定義された本来の感情の意味合いに近い生起過程の実現を可能にした。また、感情推測の式を組み込むことで、生起条件が理解しやすいよう改善された。この論理式を基に、実装の生起条件に関するプログラムの条件を変更した。実装の変更としては、あらかじめ好き嫌いや感情推測を実装していたため、生起条件の記述の変更のみである。感情生起の過程としては、図 3 における初期信念が追加された後、外部入力で信念が追加され (図 3 の a を入力として行う)、感情が生起する。

4.3 *Deserve*(ふさわしさ) を用いた論理式の形式化

先行研究 [9][10] では、*HappyFor*(ともに喜ぶ) の論理式にオペレータ *Deserve*(ふさわしさ) が条件として加えられていた。しかし、*Deserve*(ふさわしさ) は他者から *Des*(望ましさ) を伝えられたとき、自動的に追加される仕様になっていたため、本来の意味では扱われていなかった。アクションの機能の不具合により、一度 *Deserve* の条件を通さないと感情が生起しないようになっていたために追加されていたが、アクションの機能を改善したため実装としては必要がなくなった。(アクションの機能の改善については 5 章で述べる。) [9][10] では *Deserve*(ふさわしさ) の定義は明確には存在していなかったが、OCC theory では、自身が他者に対して感じるふさわしさは、*HappyFor* を含む分類④の感情の度合いに関連するものとされていた。そこで、分類④の感情の生起条件として組み込めるよう、4.1 章の二つ目の案で挙げた他者への好き嫌いをもとにした (他者の望ましい) イベントに対する自身の望ましさの意味合いを、ふさわしさとして論理式で定義することにした。

新たに形式化したオペレータ *Deserve*(ふさわしさ) の条件は、

$$Deserve_{f_{DSR}(f_L, d_j)}^{i,j} \varphi := Love_{f_L}^i(j) \wedge Bel^i Des_{d_j}^j \varphi \quad (10)$$

である。これは「エージェント i がエージェント j のことを f_L 程度好きだと感じており、かつ、 φ の成立がエージェント j にとって d_j 程度望ましい」ことであると i が信じたならば、程度 $f_{DSR}(f_L, d_j)$ の *Deserve* という信念を保持する」ということを意味する。度合いを計算する $f_{DSR}(f_L, d_j)$ は、 f_L, d_j の増加関数としている。これは、相手に好意を持っている場合に、相手の望ましいと感じるイベントをふさわしく感じるということを表している。

この表現を用いることで、4.1 章で述べた生起過程の問題点であった、自身の望ましさを条件としていた部分を、ふさわしさとして実現できる。上の式を、*HappyFor* の式に組み込んで再定義した式を以下に示す。

ふさわしさを考慮した場合、*HappyFor*(ともに喜ぶ) の生起条件は、

$$HappyFor_{f_H(f_{DSR})}^{i,j} \varphi := Bel^i Bel^j \varphi \wedge Deserve_{f_{DSR}}^{i,j} \varphi \quad (11)$$

である。これは「エージェント i が φ の成立がエージェント j に f_{DSR} 程度ふさわしいと感じており、かつ φ の成立を j が信じた」ことを i が信じたならば、程度 $f_H(f_{DSR})$ の *HappyFor* という感情を生起する」ということを意味する。度合いを計算する $f_H(f_{DSR})$ は、 f_{DSR} の増加関数としている。この論理式では、相手にふさわしいと感じるイベントが成立したときに、ふさわしさの度合いに応じてともに喜ぶということを表している。

4.4 Fortunes-of-others emotions の論理式についての議論

④Fortunes-of-others emotions の感情について、感情推測の論理式 (8) を利用した論理式 (9) と、ふさわしさの論理式 (10) を利用した論理式 (11) の二通りの生起条件を示すことができた。

論理式 (8) 及び (10) から、論理式 (9) と論理式 (11) は同じ条件を表しているため、どちらの論理式でも感情推測とふさわしさの両方が反映されていることを示すことが出来る。実装については、両方実装すれば二重の動作を行うことになってしまうため、二つの式を比較検討し、片方を実装することにした。検討した結果、論理式 (9) の方がより「他者の良い結果、悪い結果についての感情」の意味に近い表現であると判断し、④の感情の生起条件として論理式 (9) の方のみを実装することにした。

5 アクションの機能改善と不具合の修正

[9][10] の実装において、信念を伝えるアクションは、!chat と !report の二つが可能であった。この機能を使用した例を以下に示す。

エージェントを i, j とし、アクションの動作者を i とする。もし i が !chat を行ったとしたら、 i の初期信念に friend(j) がある場合、直前に i の信念に加えられた i の望ましさが j に伝えられる (j の信念に i の望ましさが追加される)。同様にして、!report を行うと、直前に i の信念に加えられた i のイベント成立を伝えられる。従来の実装の問題点として、以下のことが挙げられる。ふさわしさの不具合については、4 つめで述べる。

- アクションを行ったエージェントの初期信念プログラムに、複数の友人エージェント (friend) が設定されていた場合、最初に friend 設定されているエージェントに無条件に送られてしまうため、ある特定の一人にしか信念を伝えられない。
- 直前の外部入力を伝えるため、伝える内容を指定することができない。また、望ましさとイベント成立のみしか伝えられない。(オペレータには、イベント成立のための努力や可能性なども存在する)
- 信念として持っているものをそのまま送るため、事実を伝えることしかできない。(嘘をついたり、実際と度合いを変えて伝えることができないのは、現実で対話する状況を踏まえていない)
- !chat の機能で望ましさを伝える際、必要のない信念も余分に同時に送られていたことが判明した。それにより、ふさわしさの条件にその信念が含まれていたため、!chat を必要とする感情の生起は、一度ふさわしさを条件として参照しないと生起しない。

そこで、問題点を解消するため、アクションの実装を見直し、機能の改善を行った。アクションを !chat のみにし、伝える相手を第一引数、伝えたい信念を第二引数に取るよう実装を変更した。新たに実装し直した !chat の実行例を以下に示す。

エージェントを i, j 、イベント φ とする。 i のアクションとして入力 !chat($j, \text{des}(i, \varphi)[\text{degOfCert}(di)]$) を行った場合、 j の信念ベースに $\text{bel}(i, \text{des}(i, \varphi)[\text{degOfCert}(di)])$ が追加される (第一引数は第二引数の主語を指している)。追加された信念は「 j が i についての φ の成立が i にとって di 程度望ましい」ことの成立を信じる」という意味である。第二引数には好きにどの信念を入れることもできる。例えば努力を伝えたい場合は、!chat($j, \text{effort}(i, \varphi)[\text{degOfCert}(e)]$) と入力を行うことで「 i についての φ の成立のために i が e 程度努力したこと」が伝えられる。

これにより、!chat によって望ましさだけでなく、可能性や努力、実際に信じていない嘘の信念を伝えるこ

とも可能にした。また、不具合のあった余分な信念が送られないように実装し直し、相手エージェントの信念に、伝えられた信念のみが追加されるよう実装を改善した。

6 検証実験

本章では、本研究の妥当性を検証するために行った検証実験について述べる。実験では、新たに追加したアクションの機能や、変更した感情を含めた複数の感情生起の動作確認を行い、生起した感情とその生起過程が人間らしいかどうかを検証した。

6.1 実験方法

好き嫌いを含む複数の感情表現が描かれている文学作品を用意し、そのシナリオにそって信念を適宜保持させていくことで、感情の生起を確認する。人間らしい感情生起の検証であれば、本来、被験者を用いた心理学実験を行うことが理想であると考えられる。しかし、それには大規模な調査が必要になり、困難であるため、その代替として文学作品のストーリーを利用した感情生起の検証を行う。

本研究の検証では、ストーリー「バターはどこへ溶けた?(Where Has My Butter Gone?)」[17]を用いた。このストーリーを用いた理由としては、周囲の環境が動的に変化する中で、複数のエージェントによって自身や他者の目標達成を行う過程において感情を生起する場面が多々描かれており、望ましさや行動が明確に示されていたことが挙げられる。

まず、実験を行う前に物語を場面ごとに区切り、文中から読み取れるイベントや信念、感情の表現を抜きだし、それによってシナリオを作成した。シナリオに従い、初期信念としての他者への魅力と親密さを保持させ、信念や他者に伝える行動を追加し、感情生起を検証する。検証実験は、先行研究 [9][10] で用意されていた、TCP の 49999 番ポートに外部から接続することでエージェント内への入力可能な環境で行った。この検証は、感情の生起過程と、感情がストーリーに即したものであるかを確認するため、保持している感情の時間経過による減衰などは考慮せず、感情から起こす行動選択の機能は扱わないことにする。

6.2 実験とその結果

本章では、ストーリー「バターはどこへ溶けた?(Where Has My Butter Gone?)」[17]の中盤の場面で行った実験とその結果を紹介する。実験で使用したエージェントとイベント、シナリオは、以下の通りである。シナリオは文中の表現を使っており、要約した内容を示している。

- エージェント … キツネのマイケルとジョニー (fox), ネコのミケ (mike), 双子のリス (squirrel)
(キツネ 2 匹、リス 2 匹は同じ動作をするため、まとめて一つのエージェントとして設定した)
- イベント … バターを得る (butter), 行動したがバターが見つからなかった (no_butter),
バターのために頭を使ってうまく立ち回る (butter_brain), バターが腐る (butter_rot),
この後バターを処分する (butter_clean)
- シナリオ … キツネとネコにとって、バターはすべての幸せだと信じられていた。バターを自分たちの
実力で計画的に探し出すことをキツネたちは誇りに思っており、ネコのミケはそれに少し憧れている。
キツネは、以前バターと一緒に探したが役に立たず、見つけたバターを大幅に減らしたネコのことを良
く思っていない。キツネは努力してバターを探していると、気のいいリス達に会い、すかさず気のいい

いキツネを演じる。それによりバターのありかを聞くことに成功し、自身の行動を誇りに思う。大きなバターを目にして驚いたキツネは「バターが腐っている」と嘘を付き、リス達は申し訳なさそうにする。「処分しておく」と伝え、リスは有難がって帰っていき、キツネは高笑する。その一方で、ミケは努力していたが、疲れ果て、バターが見つからないのではないかという恐怖に苛まれる。走り回った後、再度キツネに出会い希望が見えたが、「バターはなかった」と嘘をつかれ、ミケはとぼとぼと戻る。キツネはミケをあしらって気分が良く、笑いをこらえる。

6.2.1 初期信念と起動時の結果について

実験で設定した初期信念を表 1 に、その信念によって起動時に生じた感情の結果を図 4 に示す。また、[12] で可能になったため、初期信念として保持させた望ましさや賞賛についても示している。賞賛については、「エージェントにとって、どの人が起こしても望ましい行動だと思う」という定義で実装しているため、望ましさと同様の形式の信念にしている。また、魅力と親密さの度合いやその他の信念は、ストーリーに明示的に描かれている部分と、エージェントごとの個性や以前の出来事を踏まえて任意で設定した。図 4 の「保持している信念」の魅力や親密さは最初の () 内に相手エージェント、次の () 内に度合いを表記する。また、その他の信念の最初の () 内は誰に対するイベントであるかとイベント名、次の () 内に度合いを表記する。

表 1 エージェントごとの初期信念

エージェント	設定した初期信念	実際の入力 (初期信念プログラム)
キツネ	魅力的でない (リス)(0.2) 親密さ (リス)(0.1) 魅力的でない (ミケ)(0.3) 親密さ (ミケ)(0.2) 望ましさ (キツネ butter)(0.2) 賞賛に値する (キツネ butter_brain)(0.3)	unappealing(fox,squirrel)[degOfCert(0.2)]. familiar(fox,squirrel)[degOfCert(0.1)]. unappealing(fox,mike)[degOfCert(0.3)]. familiar(fox,mike)[degOfCert(0.2)]. des(fox,butter)[degOfCert(0.2)]. des(everyone,butter_brain)[degOfCert(0.3)].
リス	魅力的である (キツネ)(0.1) 親密さ (キツネ)(0.1)	appealing(squirrel,fox)[degOfCert(0.1)]. familiar(squirrel,fox)[degOfCert(0.1)].
ミケ	魅力的である (キツネ)(0.2) 親密さ (キツネ)(0.2) 望ましさ (ミケ butter)(0.2)	appealing(mike,fox)[degOfCert(0.2)]. familiar(mike,fox)[degOfCert(0.2)]. des(mike,butter)[degOfCert(0.2)].

生じた感情の出力結果 (起動時)

```
[fox] hate(fox,squirrel)0.30000000000000004      ← キツネはリスのことが 0.3 程度嫌い
[fox] hate(fox,mike)0.5                          ← キツネはミケのことが 0.5 程度嫌い
[squirrel] love(squirrel,fox)0.2                 ← リスはキツネのことが 0.2 程度好き
[mike] love(mike,fox)0.4                         ← ミケはキツネのことが 0.4 程度好き
```

図 4 実行結果 (起動時)

6.2.2 追加した信念と生じた感情について

実験で追加した信念や行動と生じた感情の結果を、表 2、表 3 に示す。エージェントの状況は、表の途中の ♣ の後に続く文で説明し、信念追加後に生じた感情を ○ の後に示す。

表 2 キツネとリスの場面における実験結果

エージェント	追加した信念や行動	実際の入力
♣ キツネはバターを努力して探し見つかる可能性を感じており、上手く立ち回るのは困難なことだと思う		
キツネ	努力 (キツネ butter)(0.2)	fox effort(fox,butter)[degOfCert(0.2)]
キツネ	可能性 (キツネ butter)(0.2)	fox prob(fox,butter_brain)[degOfCert(0.2)]
キツネ	○ Hope(望み) 出力: [fox] hope(fox,butter)0.4	
キツネ	可能性が低い (キツネ butter_brain)(0.3)	fox prob_not(fox,butter_brain)[degOfCert(0.3)]
♣ キツネはリスにバターが望ましいことを伝え、見つけたことを伝えられ、上手く立ち回れたと感じる		
キツネ → リス	望ましさを伝える (キツネ butter)(0.2)	fox !chat(squirrel, des(fox,butter)[degOfCert(0.2)])
リス → キツネ	成立を伝える (リス butter)	squirrel !chat(fox,bel(squirrel,butter))
キツネ	成立 (キツネ butter_brain)	fox bel(fox,butter_brain)
キツネ	○ Pride(誇り) 出力: [fox] pride(fox,butter_brain)0.6	
♣ キツネはリスに連れていかれ大きなバターを見て、バターが腐っていると嘘をつく		
キツネ	成立 (キツネ butter)	fox bel(fox,butter)
キツネ	○ Joy(喜び) 出力: [fox] joy(fox,butter)0.2 ○ Satisfaction(満足) 出力: [fox] satisfaction(fox,butter)0.6	
キツネ → リス	成立を伝える (キツネ butter_rot)	fox !chat(squirrel,bel(fox,butter_rot))
リス	推測 (キツネ (望ましくない (キツネ butter_rot)))(0.2)	squirrel bel(fox, des(fox,not butter_rot)[degOfCert(0.2)])
リス	○ Guess_Distress(悲しみ推測) 出力: [squirrel] guess_distress(squirrel,fox,butter_rot)0.2 ○ SorryFor(ともに残念に思う) 出力: [squirrel]sorryfor(squirrel,fox,butter_rot)0.4	
♣ キツネはバターを処分すると伝え、リスはそれを少し困難であり同時に賞賛され望ましいことだと思う		
キツネ → リス	成立 (キツネ butter_clean)	fox !chat(squirrel,bel(fox,butter_clean))
リス	賞賛に値する (キツネ butter_clean)(0.2)	squirrel des(everyone, butter_clean)[degOfCert(0.2)]
リス	可能性が低い (キツネ butter_clean)(0.1)	squirrel prob(fox, not butter_clean)[degOfCert(0.1)]
リス	○ Admiration(賞賛) 出力:[squirrel] admiration(squirrel,fox,butter_clean)0.3	
リス	望ましい (キツネ butter_clean)(0.2)	squirrel des(fox,butter_clean)[degOfCert(0.2)]
リス	○ Joy(喜び) 出力: [squirrel] joy(fox,butter_clean)0.2 ○ Gratitude(謝意) 出力: [squirrel] gratitude(squirrel,fox,butter_clean)0.5	

表 3 ミケとキツネの場面における実験結果

エージェント	追加した信念や行動	実際の入力
♣ ミケは可能性を感じてバターを探す努力をしていたが、見つからないかもしれないと思い、結局見つからなかったなんてことにはなってほしくないと感じ、そうならないよう努力する		
ミケ	可能性 (ミケ butter)(0.2)	mike prob(mike,butter)[degOfCert(0.2)]
ミケ	○ Hope(望み) 出力: [mike] hope(mike,butter)0.4	
ミケ	努力 (ミケ butter)(0.2)	mike effort(mike,butter)[degOfCert(0.2)]
ミケ	可能性 (ミケ no_butter)(0.5)	mike prob(fox,no_butter)[degOfCert(0.5)]
ミケ	望ましくない (ミケ no_butter)(0.4)	mike des(fox,not no_butter)[degOfCert(0.4)]
ミケ	○ Fear(恐れ) 出力: [mike] fear(mike,no_butter)0.9	
ミケ	防ぐ努力 (ミケ no_butter)(0.2)	mike effort(mike,not no_butter)[degOfCert(0.2)]
♣ ミケは以前バターと一緒に探したキツネに、探している (望ましい) と伝え、なかったと嘘をつかれる		
ミケ	推測 (キツネ (望ましい (キツネ butter)))(0.2)	mike bel(fox, des(fox,butter)[degOfCert(0.2)])
ミケ → キツネ	望ましさを伝える (ミケ butter)(0.2)	mike !chat(fox, des(mike,butter)[degOfCert(0.2)])
キツネ → ミケ	不成立を伝える (キツネ butter)	fox !chat(mike,bel(fox,not butter))
ミケ	○ Guess_Distress(悲しみ推測) 出力: [mike] guess_distress(fox,butter)0.2	
ミケ	○ SorryFor(ともに残念に思う) 出力: [mike] sorryfor(fox,butter)0.6	
ミケ	不成立 (ミケ butter)	mike bel(mike,not butter)
ミケ	○ Disappointment(落胆) 出力: [mike] disappointment(mike,butter)0.6	
ミケ	成立 (ミケ no_butter)	mike bel(mike,no_butter)
ミケ	○ Distress(落胆) 出力: [fox] distress(mike,no_butter)0.4	
ミケ	○ FearConfirmed(恐れていたことが確定) 出力: [fox] pride(fox,butter_brain)1	
♣ ミケはとぼとぼ帰り、ミケはバターが見つからないことは望んでいなかったらうとキツネは思う		
キツネ	成立 (ミケ no_butter)	fox bel(mike,no_butter)
キツネ	推測 (ミケ (望ましくない (ミケ no_butter)))(0.2)	fox bel(mike, des(mike,not no_butter)[degOfCert(0.2)])
キツネ	○ Guess_Distress(悲しみ推測) 出力: [fox] guess_distress(fox,mike,no_butter)0.2	
キツネ	○ Gloating(ほくそ笑む) 出力: [fox] gloating(fox,mike,no_butter)0.7	

* 表 2, 表 3 に示している感情の出力について、計算時に出た微小な端数が出力されている場合、その端数は省略して示している。

6.3 分析と考察

図4の実行結果では、表1の初期信念により、好き嫌いの感情が度合い付きで生じたことを確認した。度合いについても、魅力と親密さのオペレータの程度の値を加算した値が、感情の程度として出力された。また、生じた感情はシナリオと矛盾していない感情であった。

表2と表3の結果について、シナリオの流れに沿って信念を追加した場合も、シナリオに即した感情生起であることが確認できた。他者に関する感情は、表2では *SorryFor*、表3では *SorryFor* と *Gloating* が生起し、自身にとっての望ましさではなく、他者にとっての結果と好意を条件として生起した。

また、実装を改善したことで、エージェントの感情生起がより豊かになった。表2の *SorryFor* は、リスが「キツネがバターを得る」こと自体の望ましさは持たないような状況であるため、以前の実装では生起しない感情であった。この *SorryFor* は、「気のいいキツネを演じていたキツネに対して、実際に気のいいリスは、少しは良い印象を持ったはずであり、その好意によってキツネの悪い結果に対してともに残念に思う」という感情生起を実現しており、自然な生起過程であると考えられる。

表3のミケのキツネに対する *Guess_Distress* と *SorryFor* は、物語に明示的に描かれていない感情である。しかし、キツネに対してともに残念に思うことは、シナリオから推測できる人間らしい感情生起であり、妥当な感情生起であると判断した。一方で、可能性として、ミケは自身のショックにより他者の感情まで気にする余裕がなかった場合も考えられる。これを踏まえ、自身の状態と他者への感情の生起との関連を考慮して、生起する感情の度合いの設定をより正確に行う必要性が感じられた。

検証で扱ったシナリオでは個性の異なるエージェントを使用したが、そのエージェントの個性に沿った感情の生起が確認できた。例えば、キツネは自身の利益にしか興味がないため他者に魅力を感じず、結果では他者に対して思いやりのない感情を生起している。逆にリスやミケは、他者に対して思いやりのある感情が生起している。これは、状況ごとに感じる望ましさなどに加えて、基から保持している信念によって感情が生起するため、より個性が反映された結果になったと考えられる。また、以前ではただ合理的に望ましさや行動をもとに感情が生起していた所を、他者への好き嫌いという一見非合理的な感情によって他の感情を生起させたため、より人間らしさを追求できた。

!chat の機能については、エージェントが伝えたい信念を直前に追加されたものでなくても伝えることが可能になり、嘘の信念も伝えることが出来た。伝えられた信念は、指定した相手エージェントの信念に保持され、その後の感情生起の条件としても利用されていることが確認された。

6.4 検証実験における課題

検証実験では、オペレータの度合いをストーリーから推測して設定しており、感情生起に必要なパラメータが明示的に描かれていない場合は、読み取れる部分から追加するかを判断した。この際、明確な指標は存在しておらず、信念を追加する者の判断に頼っているため、信念追加の正当性を示すことが困難である。また、確認する感情の度合いが適切であるかは分析していない。そのため、度合いや信念追加の設定をより適切に行い、生起する感情の分析も正確に行える方法を検討していきたい。

シナリオに描かれている感情のうち生起しなかったものとして「驚き」があった。「驚き」は反応としての感情だと考えられるが、本研究に用いている OCC theory では扱われていないため、生起させることが出来ていなかった。これは先行研究の [13] でも課題として挙げられていたが、その後の研究でも OCC theory に

よる感情の定義に即した実装を実現できた後の課題であると考えられ、従来は実装するまでに至らなかった。しかし、より人間に近い感情表現を行わせるためには、今後実装に追加する必要があると考えられる。また、「疲れ果てている」などの身体的、または精神的疲労についても反映させることが出来なかった。これについても今後の課題といえる。

シナリオでは最後に「キツネが高笑う」という行動が入っているが、これはキツネの以前に成立した「バターを得る」に対する喜びや満足、「バターのために上手く立ち回る」に対する誇りによる行動だと考えられる。また、リスを追い払うことで成立したそれらのイベントの二度目の成立とも言える。しかし、二度目に起こったイベントは別のイベントとして設定しなければならず、その設定は行わなかったため、その時点で喜びなどの感情が再度生起することはなかった。この点については二度目に起こるイベントを可能にするか検討が必要だと考えられる。また、その時「ほくそ笑む」が生起しなかったことについては、リスはキツネにバターが望ましいという態度をそこまで見せている様子がなく、キツネはリスにとって悪い結果が起こったとは思っていないため、生起しないのが妥当である。

オペレータについては、「イベントの成立が困難である」を「可能性が低い」という意味のオペレータで表現しているのは、本来の意味合いとは微妙に異なるとも考えられる。シミュレーションで信念を追加するときも妥当性が問われた。また、期待しているイベントに対する感情生起の条件として、イベント成立のための努力のオペレータがあるが、ない場合に感情が生起する場面はあると考える。そのため、現在は感情生起の度合いのみに関連するオペレータは存在していないが、なくても生起するがあれば度合いに影響を与えるようなオペレータの実装を行うことも視野に入れていきたい。

!chat の機能でも、課題が挙げられる。表 2 ではキツネが「バターが腐っている」と伝える場面でリスは疑わなかったが、現実世界ではそのまま信じず疑う場面も多々存在する。しかし、現在の !chat の機能では、嘘はつけても、伝えられた相手はそのまま信念として受け入れる仕様になっているため、疑うことが出来ない。今後現実に即した実装にするため、状況を踏まえて伝えられたことを疑うような機能も加えていく必要がある。

表 3 で扱ったシナリオでは、イベントの設定が困難であった。「行動したがバターが見つからなかった」は「バターを得る」の不成立とも捉えられるが、ミケが感じていた不安事項は、「バターがない状況(バターを得るの不成立)」の成立ではなく、劣等感を持っているミケにとって「(ただのネコである自分が)行動してもバターを結局見つけられない」ということだと考え、今回のイベント設定に至った。しかし、そのイベントは、大きく捉えて「自身の行動が結果に結びつかない」というイベントでもあると考えられる。今回はバターが関連する事象に限定して検証を行ったが、もしそのように設定した場合、ミケのバター以外の事象に対する感情の生起要因として扱えるかもしれない。このようにイベントの設定には検討が必要であり、複数の事象が生じる状況において、どの事象に焦点を当てるかは難しい課題である。

7 終わりに

本研究では、人間らしい感情を生起するエージェントの実現に向けて、Love と Hate の感情を実装に追加し、その感情を他者の望ましさについての感情の生起条件として利用することで、より自然な感情の生起過程になるよう論理式を再定義し、実装を改善した。それに加え、他者に対するふさわしさの定義を検討し、アクションの機能の改善も行った。これにより、検証実験では、エージェントの感情表現が豊かになり、OCC theory の理論に即した人間らしい生起過程で感情が生起していることが確認できた。また、初期信念の設定を可能にしたことで、エージェントごとの個性を反映する結果に繋がったと考えられる。

今後の課題として、好き嫌いの感情は初期から保持している不変的なものとして実装を行ったが、他者に対

して感じる魅力や親密さは状況に応じて変化し続けるものであり、好き嫌いの感情もそれに依りて変化していくべきである。本論文で紹介した実験では、好き嫌いが変化しない場面をシナリオに採用していたため一連の流れでシミュレート出来たが、ストーリーの序盤から全てシミュレートする場合は、魅力と親密さの変化する場面で初期信念を設定し直さなければならない。これでは自律エージェントの機能として不十分であるため、今後魅力や親密さのオペレータの値の更新処理を可能にする必要がある。また、現実世界では、好き嫌いの感情の生起要因とされている魅力や親密さの判定には、「外見」「自身への態度」「間柄」など様々な要素が関連しているため、他の要因と結びつけてより人間らしく設定する方法を定めることが必要である。また、他者に対しての感情は好き嫌いだけではないと思われるため、他者に対してより多様な感情を生起させることが望ましい。検証を行った際に考えられる課題については、6.4章で述べている。これらの課題を解決することで、より人間らしい感情を生起する自律的なエージェントの実現を目指していきたい。

8 謝辞

本研究及び本論文の作成にあたり、指導教官の新出尚之准教授、並びに博士前期課程今井先輩から丁寧なご指導、ご助言を賜りました。また、近畿大学理工学部の高田司郎先生には、研究結果に関して刺激的な議論を頂きました。心からの感謝の気持ちとお礼を申し上げたく、謝辞にかえさせていただきます。

参考文献

- [1] 徳久雅人, 岡田直之. パターン理解的手法に基づく知能エージェントの情緒生起. 情報処理学会論文誌, Vol. 39, No. 8, pp. 2440–2451, 1998.
- [2] 竹内将吾, 酒井あゆみ, 加藤昇平, 伊藤英則. 対話者好感度に基づく感性会話ロボットの感情生成モデル. 日本ロボット学会誌, Vol. 25, No. 7, pp. 1125–1133, 2007.
- [3] 河野泉, 池邊亮志, 和氣早苗, 上窪真一, 岩沢透, 西村健士. 感情表現を用いた対話システム EDS の開発 (1) ~ システム概要と感情モデル ~. 情報処理学会研究報告ヒューマンコンピュータインタラクション (HCI), Vol. 2000, No. 61, pp. 43–48, 2000.
- [4] 藤田恵, 片山寛子, 新出尚之, 高田司郎. 実世界の多様性に適応した BDI ロボットについて. 情報処理学会論文誌数理モデル化と応用, Vol. 5, No. 1, pp. 50–64, 2012.
- [5] Anand S. Rao, Munindar P. Singh, and Michael P. Georgeff. *Formal Methods in DAI: Logic-Based Representation and Reasoning*. Massachusetts Institute of Technology, 1999.
- [6] A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1998.
- [7] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming MultiAgent Systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.
- [8] Carole Adam, Andreas Herzig, and Dominique Longin. Logical formalization of OCC theory of emotions. *Synthese*, Vol. 168, No. 2, pp. 201–248, 2009.
- [9] 今井那緒. 複数の感情の同時生起および削除を取り入れたエージェントの実装. 2015 年度卒業論文, 奈良女子大学理学部情報科学科, 2016.
- [10] 向井香里. 時間経過による感情の程度の減衰を取り入れたエージェントの実現. 2015 年度卒業論文, 奈良女子大学理学部情報科学科, 2016.

- [11] 石川葉子. 感情に強さを取り入れたエージェントの実現. 2014 年度卒業論文, 奈良女子大学理学部情報科学科, 2015.
- [12] 塚本麻衣. 感情表現を持つ自律エージェント. 2017 年度卒業論文, 奈良女子大学生生活環境学部情報井環境学科生活情報通信科学コース, 2018.
- [13] 清水詩子. 感情表現を用いた行動決定を行うエージェントの実現. 2011 年度卒業論文, 奈良女子大学理学部情報科学科, 2012.
- [14] 池之内彰子. OCC theory に基づくエージェントの感情表現の改良. 2012 年度卒業論文, 奈良女子大学理学部情報科学科, 2013.
- [15] 山根瑞樹. OCC theory による感情表現を持つエージェントの実現. 2013 年度卒業論文, 奈良女子大学理学部情報科学科, 2014.
- [16] 今井那緒, 浅井沙良, 塚本麻衣, 新出尚之. OCC theory に基づくエージェントの感情表現と時間経過に関する論理モデル. 日本ソフトウェア科学会第 34 回大会 (2017 年度) 講演論文集, 2017.
- [17] Dean Ripplewood. バターはどこへ溶けた? Where Has My Butter Gone? 道出版, 2007.