OCC theoryに基づくエージェントの感情表現の改良

奈良女子大学 理学部 情報科学科 4回生 09251534
新出研究室 池之内彰子

概 要

近年、エージェントと人との対話を実現させる上で、感情を持ち、その感情表現を行動決定にいかせるといった、より人間に近いエージェントが近年必要とされている。従来、OCC theoryと呼ばれる感情をモデル化した理論を用いて、感情を生起し、その感情に伴った行動を選択できるBDIエージェントの実現を目指した研究がなされてきた。しかし、従来の研究では全ての感情を実現するには至っていない。

そこで本研究では、今まで実装できていなかった感情表現の実装を進め、実装した感情表現の検証と今後の問題点の議論を行った。

1 はじめに

近年、エージェントと人との対話を実現させる上で、感情を持ちその感情表現を行動決定にいかせるといった、より人間に近いエージェントが近年必要とされている。特に、癒しなどを目的に人とコミュニケーションするエージェントは、実世界において感情を持ちつつ行動できることが望まれる。

実世界のような動的環境において問題解決のために行動できるエージェントの実現には、BDIアーキテクチャが有効であると示されている [7]。BDIアーキテクチャとは、BDIモデル [6] による行為決定方式を計算機上で実現したもので、これによりBDIエージェントという人間の合理的な思考に基づいた行動を取るエージェントを構築することが可能である。BDIモデルとは、信念 (Blief)、願望 (Desire)、意図 (Intention) の3つの心的パラメータを用いた人間の合理的な行動をモデル化したものであり、BDI logicという論理モデルを持つ。

一方、OCC theory[5] と呼ばれる、心理学的見地に基づく22種類の感情をモデル化した理論が感情の形式化に多く用いられている。このOCC theoryは信念や願望などの心的状態を用いて感情の特徴付けが明確であり、また、その感情の特徴付けを論理モデルで表現可能という特徴を持つため、BDI logicを持つBDIモデルとは親和性が高く、既存のBDIアーキテクチャの実装であるJason[4]上で信念ベースを用いて実現が可能である。

上記のBDIエージェントとOCC theoryを用いて、感情を生起し自力でその感情に伴った行動が選択できるエージェントの実現を目指した研究 [1] があるが、全ての感情を実現するには至っていない。特に、「他者の感情が自分の感情生起に影響するような感情」と、「行動に対する賞賛度に焦点を当てた感情」が未実装であった。

そこで本研究では、エージェント間で通信することにより他エージェントの感情を自分の信念をして取得することで、「他者の感情が自分の感情生起に影響するような感情」を、行動を表すプランの前後でその感情が生起されたかどうか検査することで「行動に対する賞賛度に焦点を当てた感情」を新たに実装した。

BDIモデルとOCC theoryを用いてエージェントの感情を扱っている関連研究としては [1] の他に [2] や [3] がある。

Adam ら [2] は、BDI モデルでの形式化に用いられる論理体系である BDI logic に対し、「不確実だが起こると期待されている事柄」を表現する等いくらかの新たなオペレータを導入して拡張し、これを用いて、OCC theory で扱われる 22 種類の感情の特徴付けを論理式として形式化することで BDI モデルに取り込んでいる。

ただし、この研究におけるモデルではひとつの行動がひとつの感情を生起されるようになっており、複合的な感情の生起については扱えない。また、感情の度合いについても扱っていない。加えて、現段階の実装としては、状況からの感情の生起を推論する部分を独自に実現したものはあるが、他のシステムと結合できるよう汎用的にまとめられたものはなく、感情表現をエージェントの行動決定の一部として取り込めるようにはなっていないため、BDI エージェント部分の実装を一般利用者が行えない。

本研究では基本的にこの Adam らの形式化を踏まえつつ、Jason による実装を独立に進めた。これによってエージェントシステムに感情表現を組み込む汎用的な手法として利用できるようにした。

2 OCC theory

OCC thory[5] とは、Ortony, Clore, Collins らが提唱した、心理学的見地を基にした 22 種類の感情のタイプをモデル化した理論である。人間の包括的な感情を形式化しており、感情の特徴付けが明確であるため、比較的理解しやすく、計算機科学の分野において広く用いられている。

22 種類の感情をまとめると以下のようになる。

1. 事象の結果の望ましさにのみ焦点を当てたもの
   - (a) 結果の望ましさに関して（自分にとって）
     - i. 起こった事象に関するもの
       — Joy(喜び), Distress(嘆き)
     - ii. 予想に関するもの
       - 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ことに関して
         — Satisfaction(満足), FairConfirmed(恐れていたことが確定)
       - C. 予想していたことが起こらなかったことに関して
         — Relief(安堵), Disappointment(落胆)
   - (b) 結果の望ましさに関して（他者にとって）
     - i. 他者が良い結果を得た場合
       — HappyFor(共に喜ばしく思う), Resentment(憤り)
     - ii. 他者が悪い結果を得た場合
       — SorryFor(共に残念に思う), Gloating(ほくそ笑む)
2. 行動に対する賞賛度にのみ焦点を当てたもの
   - (a) 自分の行動に関するもの
     — Pride(自尊心), Shame(羞恥心)
   - (a′) Joy と Distress との混合型
     — Gratification(満足), Remorse(後悔)

(b) 他者の行動に関するもの
— Admiration(賞賛), Reproach(非難)

(b′) Joy と Distress との混合型
— Gratitude(謝意), Anger(怒り)

3. 対象物に対する好き嫌いにのみ焦点を当てたもの
— Love(好き), Hate(嫌い)

3 従来研究

1 章で述べた通り、従来研究 [1] では、全ての感情を実装するに至っていない。本章では実装されている感情の実装方法と、実装されていない感情については、実装する上での問題点を述べる。

3.1 実装に用いられた形式化

Adam らの形式化では、感情表現の生起条件を BDI logic での論理式で表現する。
① 1(a)i : Joy, Distress
② 1(a)iiA : Hope, Fear
③ 1(a)iiB : Satisfaction, FearConfirmed, 1.(a)iiC : Relief, Disappointment
④ 1(b) : HappyFor, Resentment, SorryFor, Gloating
⑤ 2 : Pride, Shame, Gratification, Remorse, Admiration, Reproach, Gratitude, Anger
⑥ 3 : Love, Hate
以上のように 22 種類の感情を 2 章の分類での 6 種類に分けると、それぞれ生起条件が同じ形で出てくる。

本節では、生起条件が同じ形のものについては、それぞれの中から 1 つの感情について、従来研究で実装されていた感情 (上記の ①〜③) の実装方法について述べる。

3.1.1 *Joy*

Joy(喜び) の生起条件は

$$Joy_i\varphi \stackrel{def}{=} Bel_i\varphi \wedge Des_i\varphi$$

と記述される。

ここで、$Bel_i\varphi$ は「i は事象 φ の成立を信じている」$Des_i\varphi$ は「i にとって事象 φ は望ましい」という意味で、上記の Joy の生起条件は「エージェント i が、事象 φ の成立を信じ、かつそれが i にとって望ましいことであるならば、i は φ の発生に対して喜びという感情を生起する」という意味になる。基本的には、この式の右辺が成立するかどうかを Jason で表現し、成立すれば Joy という感情が生起されたとする。

$Bel_i\varphi$ の表現は Jason での信念ベースをそのまま使用できるが、ここでの $Des_i\varphi$ は BDI モデルでの Desire(願望) とは別物で、Desirability(i にとって φ は望ましい) を表すため、Jason の願望つまり目標にあたる Goal を用いて表現することができない。

Adam らの形式化では、

$$Des_i\varphi \Leftrightarrow Bel_iDes_i\varphi$$

が成り立つので、$Des_i\varphi$ であることと i がこれを信念に持つことは同等となり、$Des_i\varphi$ を表現する場合、i の信念ベースにこれを追加すればよい。従って、実装は、信念ベースに φ と $Des_i\varphi$ があるかどうかを検査し、条件が揃えば $Joy_i\varphi$ を結論する、という実装方針を取っている。

3.1.2 *Hope*

$Hope$(望ましい) の生起条件は

$$Hope_i\varphi \stackrel{def}{=} Expect_i\varphi \wedge Des_i\varphi$$

と記述される。これは「i が事象 φ が成立しそうだと思っており、かつそれが i にとって望ましいことであるならば i は φ の発生に対して望ましいという感情を生起する」という意味になる。

$Expect_i\varphi$ は $Prob_i\varphi \wedge \neg Bel_i\varphi$ の略記として定義され、$Prob_i\varphi$ は「φ である可能性が高い」という意味である。従って、$Expect_i\varphi$ は「φ を完全には信じていないが、ありそうであると思っている」という意味になる。

$Prob$ は確度の低い信念をみなせるので、Jason では annotation 付き信念で表現されており、あとは Joy と同様に実装されている。

3.1.3 *Satisfaction*

$Satisfaction$(満足) の生起条件は

$$Satisfaction_i\varphi \stackrel{def}{=} Bel_i\mathbf{P}Expect_i\varphi \wedge Des_i\varphi \wedge Bel_i\varphi$$

と記述される。これは「i が事象 φ の成立を過去のあるときに期待し、かつ現在その成立を信じ、それが i にとって望ましいなら、i は φ が達成したということに対し満足している」という意味になる ($\mathbf{P}$ は「過去のある時」を表す)。しかし、この式を実装するには $Bel_i\mathbf{P}Expect_i\varphi$ の実現が難しいため、

$$Expect_i\varphi \wedge Des_i\varphi \supset \mathbf{A}(Satisfaction_i\varphi\mathbf{N}Bel_i\varphi)$$

とパラフレーズしている。これは「i が φ の成立を期待し、それが i にとって望ましいなら、次に i がその成立を信じたときに i は満足する」という意味となる ($\mathbf{A}(\psi\mathbf{N}\varphi)$ は「現時刻以後最初に φ が成り立ったときに ψ も成り立つ」を表す)。Adam らの形式化では、Des は同じエージェント i に対しては時間変化によって変わらないものとされている ($Des_i\varphi \Leftrightarrow \mathbf{G}Des_i\varphi$ が成り立つ。ここで $\mathbf{G}$ は「未来永劫に」を表す) ため、$Des_i\varphi$ を検査するタイミングはいつでもよいとされている。

そして、$Expect_i\varphi$ を結論したらしばらくそれを保持しておき、φ が信念に入ったら $Satisfaction_i\varphi$ を結論するとともに $Expect_i\varphi$ を削除する、という実装方針が取られている。

3.2 実装上の問題点

3.2.1 他者の感情が自分の感情生起に影響を及ぼす感情

従来研究では、他者の感情が自分の感情生起に影響を及ぼす感情 (3.1 の ④) が未実装であった。

例えば、$HappyFor$(共に喜ばしく思う) の生起条件は

$$HappyFor_{i,j}\varphi \stackrel{def}{=} Bel_i\varphi \wedge Bel_iDes_j\varphi \wedge Des_iBel_j\varphi$$

と記述される。

この生起条件の意味は、「『エージェント i にとって “エージェント j が事象 φ の成立を信じる” ことは望ましい ($Bel_iDes_j\varphi$)』かつ、『エージェント i は “エージェント j にとって事象 φ が望ましい” と信じている ($Des_iBel_j\varphi$)』かつ、『エージェント i は事象 φ の成立を信じている ($Bel_i\varphi$)』」となる。

これは、i, j それぞれの Bel, Des に加え、Jason にエージェント間での通信があってそれらを伝えることが可能であれば実現できるとされていたが、ここで問題となっていたのが、他エージェントとの通信である。

他者の感情が自分の感情に影響を及ぼすということは、他エージェントの感情を知る必要がある。しかし、Jason では各エージェントの心的状態はそのエージェント内でのみアクセスでき、他エージェントから知ることはできないため、他エージェントの感情に依存するような感情を実装するには、他エージェントの信念が変わる度にそれをこちらに知らせることが必要になる。他エージェントの信念を取得すること自体は、エージェント間の通信で実現できるが、“他者の感情を知るのに必要な情報” のみを “信念が変わる度に” 知らせるのをどのように実装するかが問題となっていた。

3.2.2 行動に対する賞賛度に焦点を当てた感情

従来研究では、3.2.1 の他に、行動に対する賞賛度に焦点を当てた感情 (3.1 の ⑤) が未実装であった。

例えば、$Pride$(自尊心) の生起条件は

$$Pride_i(i:\alpha,\varphi) \stackrel{def}{=} Bel_i Done_{i:\alpha}(Idl_i Happens_{i:\alpha}\varphi \wedge Prob_i After_{i:\alpha}\neg\varphi) \wedge Bel_i\varphi$$

と記述される。

ここで、$Done_{i:\alpha}$ は「i がアクション α を実行する前は φ が成立していた」、$Idl_i\varphi$ は「φ が成立することは i にとって理想的である」、$Happens_{i:\alpha}\varphi$ は「i がアクション α を実行すると φ が成立する可能性がある」、$After_{i:\alpha}\varphi$ は「i がアクション α を実行すると φ が成立する」という意味である。

従って、$Pride$ の生起条件の意味は、「『エージェント i がアクション α を実行して、事象 φ が成立することは i にとって理想的であり、かつ、i が α を実行すると φ が成立しない可能性が高いと思われていた $(Bel_i Done_{i:\alpha}(Idl_i Happens_{i:\alpha}\varphi \wedge Prob_i After_{i:\alpha}\neg\varphi))$』かつ、『エージェント i は事象 φ の成立を信じている $(Bel_i\varphi)$』」となり、大まかには「達成が困難だと思われていたことを達成することでエージェント i が事象 φ の達成を誇りに思う」という意味になる。

従来研究では、$Idl_i Happens_{i:\alpha}\varphi \wedge Prob_i After_{i:\alpha}\neg\varphi$ の部分の扱いが特に困難とされていた。Adam らの形式化では、Idl は $Idl_i Happens_{i:\alpha}\varphi$ の形でしか現れないので、これを i の信念ベースに $Idl\_Happens(\alpha,\varphi)$ の形で持っておけば、別途 Idl オペレータを考慮する必要がなく、同等の情報が持て、$Idl_i Happens_{i:\alpha}\varphi$ と $Prob_i After_{i:\alpha}\neg\varphi$ 両方が得られれば $Idl_i Happens_{i:\alpha}\varphi \wedge Prob_i After_{i:\alpha}\neg\varphi$ が結論できるとされていた。

しかし、$Happens_{i:\alpha}\varphi$ と $After_{i:\alpha}$ はアクション α を実行した後の未来に関する予見の形の信念なので、これらをいかに取得するかが問題となっていた。

4 実装方法

3 章で述べたように、従来研究 [1] ではいくつかの感情が未実装であった。本章では、未実装の感情の実現のため、その解決方法と実装方針について述べる。

4.1 実装に用いる形式化

実装に用いる形式化については、3 章で述べた 3.2.1 と 3.2.2 を用いた。生起条件が同じ形のものについては、それぞれの中から 1 つの感情について、本研究で実装した感情 (3.1 の ④, ⑤) の実装方法について述べる。

4.1.1 *HappyFor*

$HappyFor$(共に喜ばしく思う) の生起条件は

$$HappyFor_{i,j}\varphi \stackrel{def}{=} Bel_i\varphi \wedge Bel_iDes_j\varphi \wedge Des_iBel_j\varphi$$

と記述され、の生起条件の意味は、3.2.1 に記述したように「『エージェント i にとって “エージェント j が事象 φ の成立を信じる” ことは望ましい $(Bel_iDes_j\varphi)$』かつ、『エージェント i は “エージェント j にとって事象 φ が望ましい” と信じている $(Des_iBel_j\varphi)$』かつ、『エージェント i は事象 φ の成立を信じている $(Bel_i\varphi)$』」である。

この感情を実現するには、エージェント j の信念をエージェント間で通信することにより、エージェント i の信念に加えることで実現できる。

感情を生起する際の時間的経過は図 1 のようになる。

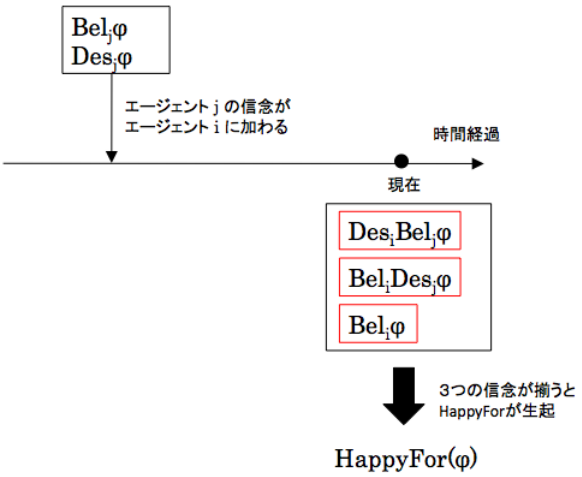

図 1: *HappyFor* の生起過程

4.1.2 *Pride*

$Pride$(自尊心) の生起条件は、

$$Pride_i(i:\alpha,\varphi) \stackrel{def}{=} Bel_iDone_{i:\alpha}(Idl_iHappens_{i:\alpha}\varphi \wedge Prob_iAfter_{i:\alpha}\neg\varphi) \wedge Bel_i\varphi$$

と記述され、の生起条件の意味は、3.2.2 でも記述した通り、「『エージェント i がアクション α を実行して、事象 φ が成立することは i にとって理想的であり、かつ、i が α を実行すると φ が成立しない可能性が高いと思われていた $(Bel_iDone_{i:\alpha}(Idl_iHappens_{i:\alpha}\varphi \wedge Prob_iAfter_{i:\alpha}\neg\varphi))$』かつ、『エージェント i は事象 φ の成立を信じている $(Bel_i\varphi)$』」である。

この感情を実現するためには、「$Idl_iHappens_{i:\alpha}\varphi \wedge Prob_iAfter_{i:\alpha}\neg\varphi$ が得られ、かつその後アクション α を実行して信念 φ が得られれば $Pride_i(i:\alpha,\varphi)$ を結論する」とすればよい。

感情を生起する際の時間的経過は図 2 のようになる。

4.2 実装方針

OCC theory における感情の生起条件を BDI logic の論理式で Adam らが形式化したものを Jason の規則として記述し、プランの前提条件として使用できるようにし、感情生起の実現と感情を行動決定に用いることができるように実装した。

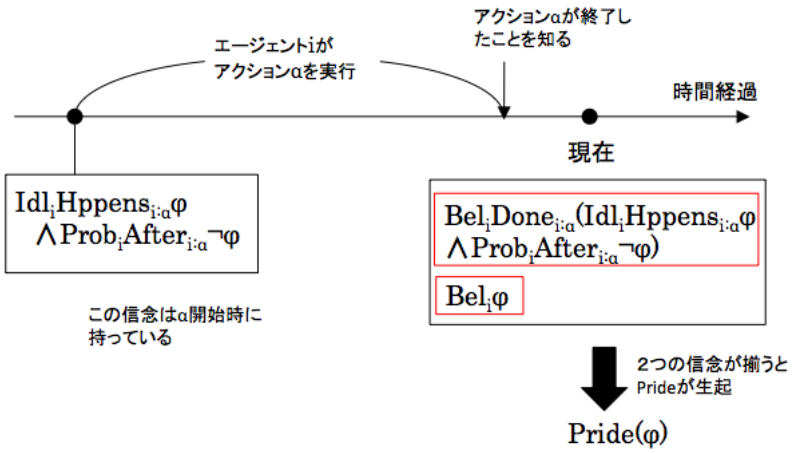

図 2: *Pride* の生起過程

4.2.1 他者の感情が自分の感情生起に影響を及ぼす感情

3.2.1 で述べたように、他者の感情に影響されるような感情の実装においては、他のエージェントの感情を知る必要があるが、Jason では各エージェントの心的状態はそのエージェント内でのみアクセスでき、他エージェントから知ることはできない。他エージェントの感情に依存するような感情を実装するには、他エージェントの信念が変わる度にそれをこちらに知らせる必要がある。

そこで、複数のエージェント間で他エージェントの感情を自分の信念として取得し、これを用いて自分の感情を生起できるように実装した。複数のエージェント間の通信方法としては、Jason の内部アクションを使用し、特定の信念について、信念追加イベントが発生した際に、同時に他エージェントに信念を送るようにして、その信念が他者に伝わるように実装した (図 3)。

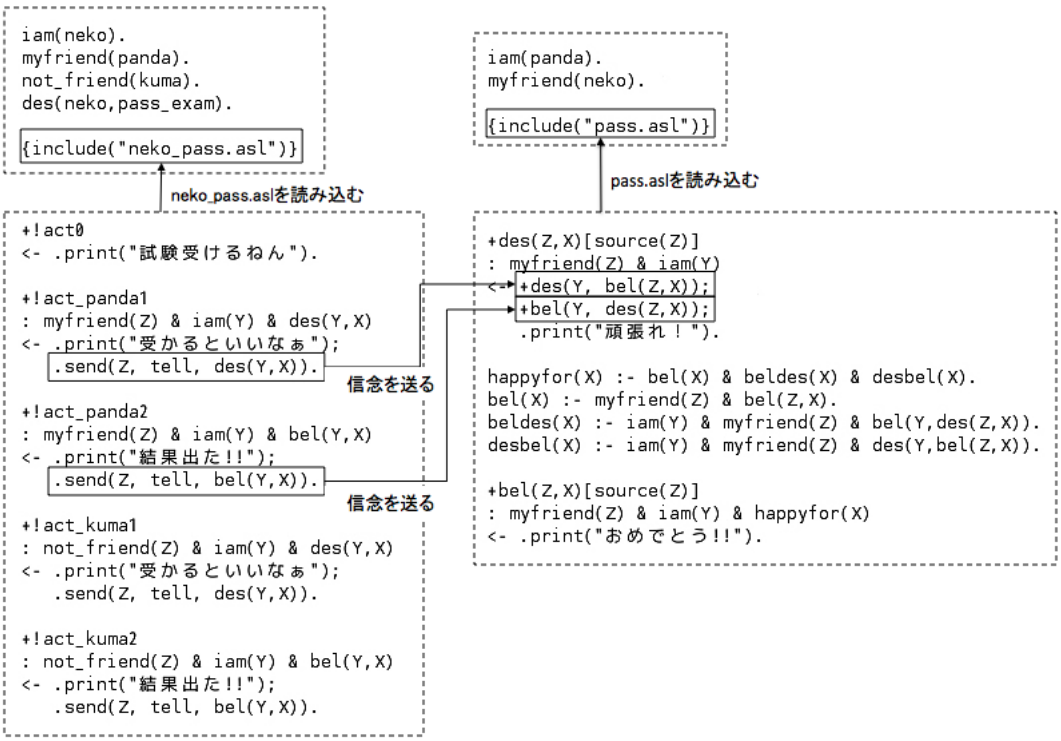

図 3: エージェント間での通信

4.2.2　行動に対する賞賛度に焦点を当てた感情

行動に対する賞賛度に焦点を当てた感情の実装においては、4.1.2 の形式化の式の $Idl_i Happens_{i:\alpha}\varphi \wedge Prob_i After_{i:\alpha}\neg\varphi$ の部分を、感情の生起に関する行動を表すプランの前後にそのプランが実行されたかどうか検査し、生起条件が揃っていれば感情を生起し、それに伴った行動をとるように実装した (図 4)。

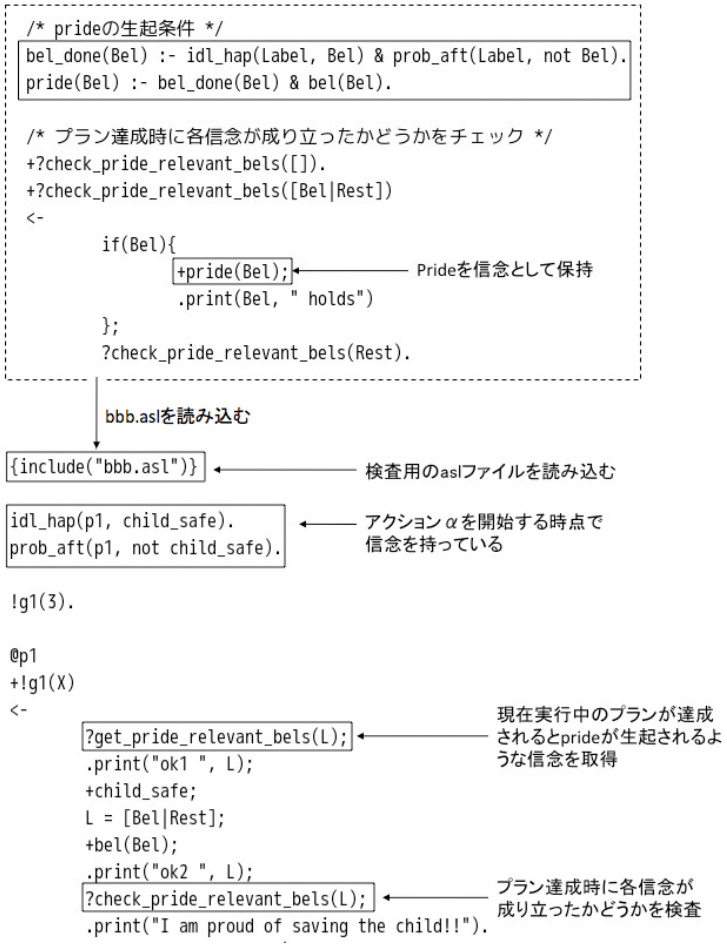

図 4: プランの検査

5　実験と結果

本章では、実装した感情が期待通りに生起され、エージェントのプランの選択に影響を与えることを実験で示す。

5.1　実験

サンプルとして、複数のシナリオを作り、そのシナリオ通りにエージェントを動かすことで、シナリオに沿った感情を生起させ、生起された感情によって行動を決定するようなプランを Jason 上に記述し、期待通りの感情生起と行動が起こることを確認した。

またそれに伴い、エージェントに対し外部から任意のタイミングで信念の追加を行うイベントや、ゴールの追加を行うイベントを与える必要があるので、外部から TCP の 49999 番ポートに接続すると、指定したエージェントにこれらのイベントを与えられるような環境を用意した。

5.2 結果

5.2.1 *HappyFor*

共に喜ばしく思う *HappyFor* を生起させるシナリオとしては、以下のように設定する。

エージェント panda と neko がいて、この 2 人は友達同士だとする。neko が試験を受け、試験に受かった (pass_exam) ことを知った panda は友人が試験に受かったことを共に喜ばしく思い、neko に対して「おめでとう」と言葉をかける。

このシナリオは図 5 のようなプラン (一部分) で実装される。neko の信念を受け取り、ゴール act_panda2 が追加された際に、*HappyFor* を生起していれば図 5 の上のプランが選ばれ、生起されていなければ下のプランが選ばれる。

操作としては図 6 のように実行し、エージェント neko に試験に受かったという信念 pass_exam を追加し、ゴール act_panda2 を追加する。そして、neko の信念を受け取ったエージェント panda が *HappyFor* を生起していたら、共に喜ばしく思うという旨の言葉をかける。結果としては、図 7 のようになり、期待通りの感情生起と行動決定を取ることができた。

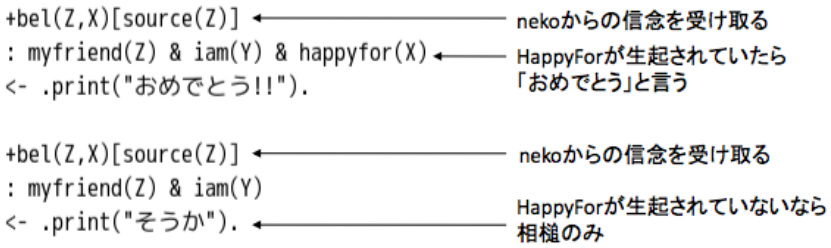

図 5: *HappyFor* を生起させるプラン

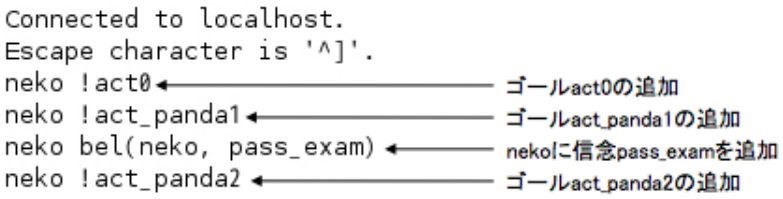

図 6: *HappyFor* を生起させる操作

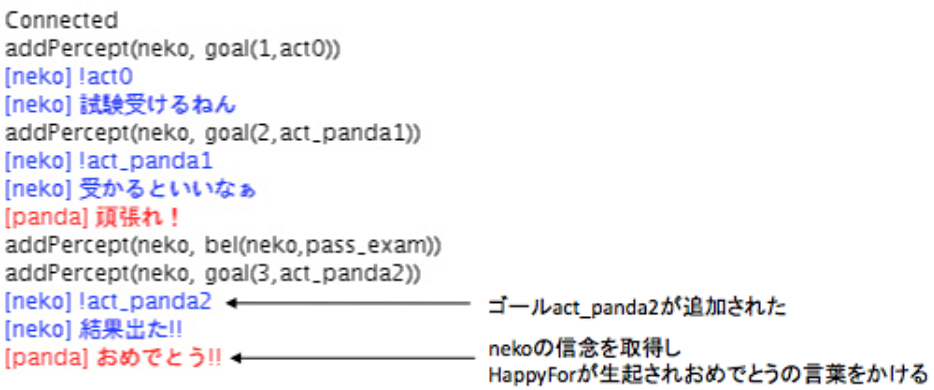

図 7: *HappyFor* を生起させた結果

5.2.2 *Resentment*

憤り *Resentment* を生起さるシナリオとしては以下のように設定する。

エージェント kuma と neko がいて、この 2 人は仲が悪いこととする。neko が試験を受け、試験に受かった (pass_exam) ことを知った kuma は知り合いが試験に受かったことに対し憤りを感じ、独り言を言う。

このシナリオは図 8 のようなプラン (一部分) で実装される。neko の信念を受け取り、ゴール act_kuma2 が追加された際に、*Resentment* を生起していれば図 8 の上のプランが選ばれ、生起されていなければ下のプランが選ばれる。

操作としては図 9 のように実行し、エージェント neko に試験に受かったという信念 pass_exam を追加し、ゴール act_kuma2 を追加する。そして、neko の信念を受け取ったエージェント kuma が *Resentment* を生起していたら、独り言を言う。結果としては、図 10 のようになり、期待通りの感情生起と行動決定を取ることができた。

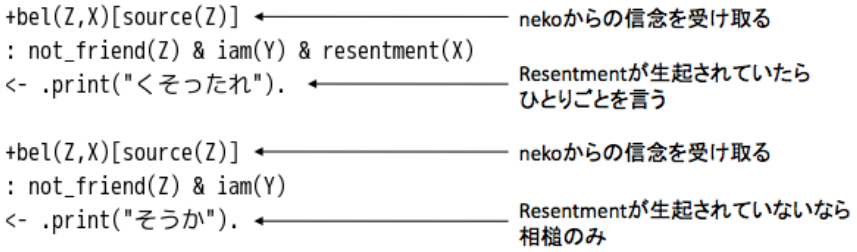

図 8: *Resentment* を生起させるプラン

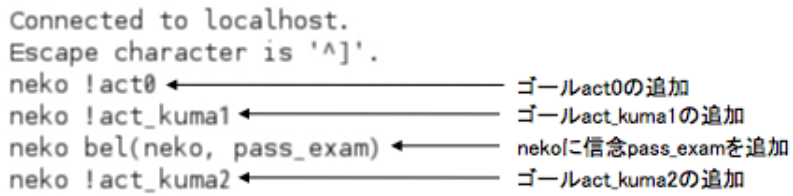

図 9: *Resentment* を生起させる操作

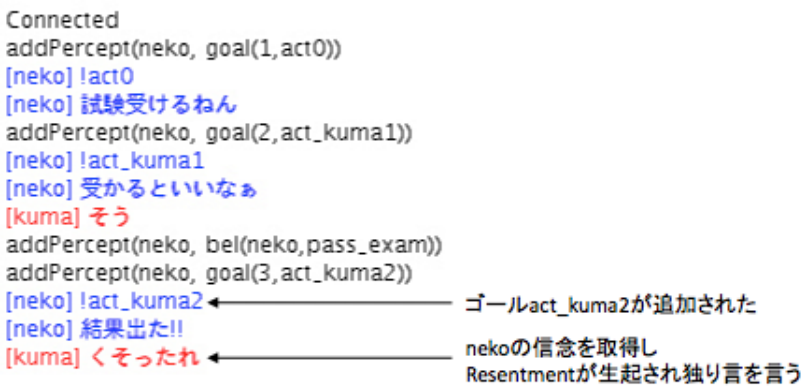

図 10: *Resentment* を生起させた結果

5.2.3 *SorryFor*

共に残念に思う *SorryFor* を生起させるシナリオとしては、以下のように設定する。

エージェント panda と neko がいて、この 2 人は友達同士だとする。neko が試験を受け、試験に落ちた (fail_exam) ことを知った panda は友人が試験に落ちたことを共に残念に思い、neko に対して「残念だったね」と言葉をかける。

このシナリオは図 11 のようなプラン (一部分) で実装される。neko の信念を受け取り、ゴール act_panda2 が追加された際に、*SorryFor* を生起していれば図 11 の上のプランが選ばれ、生起されていなければ下のプランが選ばれる。

操作としては図 12 のように実行し、エージェント neko に試験に落ちたという信念 fail_exam を追加し、ゴール act_panda2 を追加する。そして、neko の信念を受け取ったエージェント panda が *SorryFor* を生起していたら、自分も残念に思うという旨の言葉をかける。結果としては、図 13 のようになり、期待通りの感情生起と行動決定を取ることができた。

```
+bel(Z,X)[source(Z)]                    ← nekoからの信念を受け取る
: myfriend(Z) & iam(Y) & sorryfor(X)    ← SorryForが生起されていたら
<- .print("残念だったね...").                「残念だったね」と言う

+bel(Z,X)[source(Z)]                    ← nekoからの信念を受け取る
: myfriend(Z) & iam(Y)
<- .print("そうか").                     ← SorryForが生起されていないなら
                                            相槌のみ
```

図 11: *SorryFor* を生起させるプラン

```
Connected to localhost.
Escape character is '^]'.
neko !act0                              ← ゴールact0の追加
neko !act_panda1                        ← ゴールact_panda1の追加
neko bel(neko, fail_exam)               ← nekoに信念fail_examを追加
neko !act_panda2                        ← ゴールact_panda2の追加
```

図 12: *SorryFor* を生起させる操作

```
Connected
addPercept(neko, goal(1,act0))
[neko] !act0
[neko] 試験受けるねん
addPercept(neko, goal(2,act_panda1))
[neko] !act_panda1
[neko] 受かるかなぁ...
[panda] 頑張ってね
addPercept(neko, bel(neko,fail_exam))
addPercept(neko, goal(3,act_panda2))
[neko] !act_panda2                      ← ゴールact_panda2が追加された
[neko] 結果出た...
[panda] 残念だったね...                  ← nekoの信念を取得し
                                            SorryForが生起され残念だったと言葉をかける
```

図 13: *SorryFor* を生起させた結果

5.2.4 *Gloating*

ほくそ笑む *Gloating* を生起させるシナリオとしては以下のように設定する。

エージェント kuma と neko がいて、この 2 人は仲が悪いこととする。neko が試験を受け、試験に落ちた (fail_exam) ことを知った kuma は知り合いがが試験に落ちたことに対しほくそ笑み、「ざまあみろ」と独り言を言う。

このシナリオは図 14 のようなプラン (一部分) で実装される。neko の信念を受け取り、ゴール act_kuma2 が追加された際に、*Gloating* を生起していれば図 14 の上のプランが選ばれ、生起されていなければ下のプランが選ばれる。

操作としては図 15 のように実行し、エージェント neko に試験に落ちたという信念 fail_exam を追加し、ゴール act_kuma2 を追加する。そして、neko の信念を受け取ったエージェント kuma が *Gloating* を生起していたら、独り言を言う。結果としては、図 16 のようになり、期待通りの感情生起と行動決定を取ることができた。

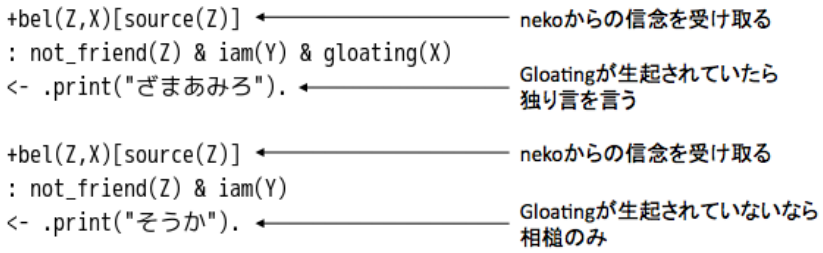

図 14: *Gloating* を生起させるプラン

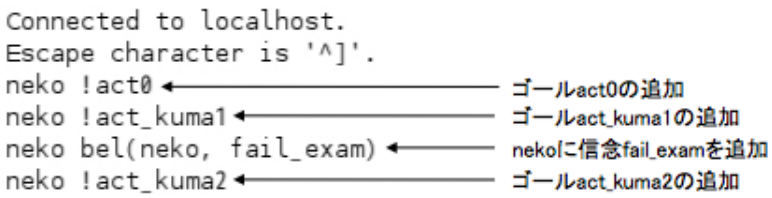

図 15: *Gloating* を生起させる操作

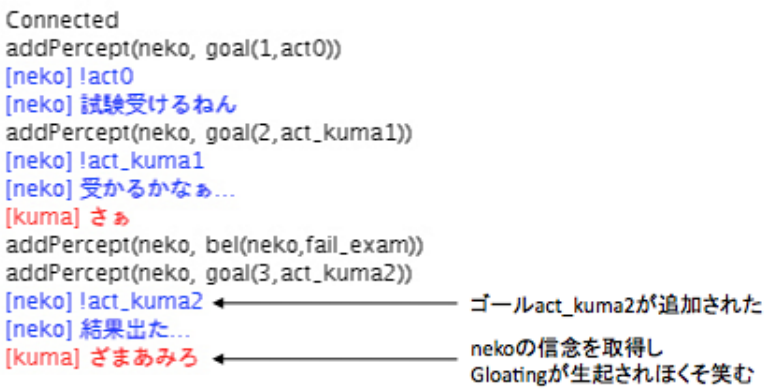

図 16: *Gloating* を生起させた結果

5.2.5 *Pride*

自尊心 *Pride* を生起させるシナリオとしては以下のように設定する。

エージェントが散歩をしているところ、川で溺れている子どもを発見し、エージェントは子どもを助けるという行動をとる。行動が終了した際に、子どもが無事 (`child_safe`) という信念を得たら、信念の達成を誇らしく思い、人に報告する。子どもが無事 (`child_safe`) という信念を得られなかったら、*Pride* は生起されず、その場を立ち去る。

このシナリオは図 17 のようなプラン (一部分) で実装される。子どもが無事 (`child_safe`) という信念があり、*Pride* が生起されていれば、子どもを助けたことを誇りに思い、人に報告しにいく。

結果としては図 18 のようになり、期待通りの感情生起と行動決定を取ることができた。

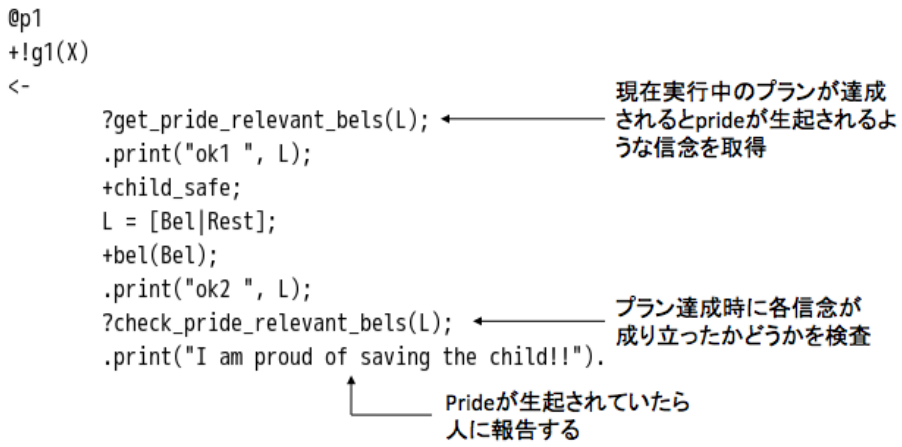

図 17: *Pride* を生起させるプラン

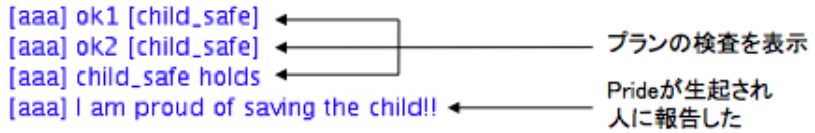

図 18: *Pride* を生起させた結果

6 終わりに

以上の結果より、ある特定のシナリオにおいては、エージェントは OCC theory に基づいて形式化された感情を推論して生起し、またそれに従った行動を取ることができ、従来の研究では実現されていなかった新たな感情を実現した。

今後の課題としては、本研究で実装した感情の妥当性の評価がある。本研究では、3.1 に述べた Adam らによる形式化をそのまま使用しているため、今回実現した感情の妥当性も Adam らの形式化に依存することになるが、その妥当性も含めて今後議論する必要がある。

また、今回実装した「行動に対する賞賛度に焦点を当てた感情」では、感情の生起に関する行動を表すプランの前後に、そのプランが実行されたかどうかを検査するプランを別に記述したが、これよりも効率のよいプランの検査ができる方法を再考慮する必要がある。

そして、本研究で実装に至らなかった感情の実現を進める必要があり、それに伴い問題となってくる、複数の感情の競合をどのように処理するか、対立関係にある感情の同時発生の扱いや、複合的な感情をどのように定義するかを解決することで、より人間に近いエージェントの実現を目指す。

謝辞

本論文の執筆及び研究にあたり、指導教官の新出尚之准教授にはいつも丁寧なご指導と助言を賜りました。心から感謝の気持ちとお礼を申し上げたく、謝辞にさせていただきます。

参考文献

[1] 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. 奈良女子大学理学部情報科学科 2011 年度卒業論文, 2012.

[2] Carole Adam, Andreas Herzig, and Dominique Longin. A logical formalization of OCC theory of emotions. *Synthese*, Vol. 168, No.2, pp.201-248, 2009.

[3] Patricia Augustin Jaques and Rosa Maria Vicari. A BDI approach to infer student's emotions in an intelligent learning enviroment. *Comput. Educ.*, Vol.49, pp.360-384,September 2007.

[4] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. Programming Multi-Agent Systems in AgentSpeak using Jason. John Wiley & Sons, 2007.

[5] A.Ortony, G.L. Clore, and A Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.

[6] Munindar P. Singh, Anand S. Rao, and Michael P. Georgeff. Formal method in DAI. *Multi-agent Systems: a Modern, Approach to Distributed Artificial Intelligence*, 1999.

[7] 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, 2012.