感情に程度の強さを取り入れたエージェントの実現

奈良女子大学 理学部 情報科学科 4 回生 11251452
新出研究室 石川葉子

概 要

ロボットに人間らしい感情表現を付加することは，人間との対話を円滑にするため重要である．特に，生起した感情に基づき，自ら行動決定できることが求められている．本研究では，OCC theory と BDI エージェントを組み合わせた Adam らの論理体系をもとに実装を行う．先行研究によって定義された，程度の強さを取り入れた感情の形式化をもとに，より人間らしい感情表現を目指す．また，生起した感情の削除を可能にすることで，感情を連続して生起できるようにした．検証実験から，エージェントが度合い付きの感情を生起し，感情やその度合いによって行動選択することを確認した．最後に，実装した感情表現の妥当性の検討と，より人間らしさを求めるため必要な課題点を述べる．

1 はじめに

ロボットに人間らしい感情表現を付加することは，人間との対話を円滑にするため重要である．近年，日常生活の中で人間と関わりながら活動する社会的ロボットの開発が進んでいる．人間がロボットを対等なパートナーとして認識するためには，より人間らしいエージェントが望まれている．特に，生起した感情に基づき，自ら行動決定できるエージェントが求められている．

周囲の環境が常に変化する実世界において，問題解決のため行動するロボットの実現として BDI アーキテクチャが有効である [8]．BDI アーキテクチャとは，BDI モデル [4] による行為決定方式を計算機上で実現したものであり，これにより BDI エージェントという人間の合理的思考に基づいた行動をと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，人間の行動を信念 (Belief)，願望 (Desire)，意図 (Intention) の 3 つの心的状態でモデル化したものである．このモデルは BDI logic という論理モデルを持つ．

人間の感情を分類する理論に OCC theory[3] がある．OCC theory では，信念や願望などの心的状態を用いて人間の感情を分類，特徴付けている．この感情の特徴付けは論理モデルで表現可能であり，BDI logic を持つ BDI モデルとは親和性が良い．実際に，OCC theory によって分類された感情は，既存の BDI アーキテクチャの実装である Jason[2] 上の信念ベースを用いて実現が可能である．

Adam ら [1] は，OCC theory の理論を BDI モデルに取り込むことで，感情を論理式でモデル化した．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象」など，新たに複数のオペレータを導入している．これらを用いることで，OCC theory で扱われている 20 種類の感情を論理式として形式化し，BDI モデルに取り込んでいる．しかし，この研究におけるモデルでは 1 つの行動につき 1 つの感情しか生起できず，複合的な感情については扱っていない．加えて，感情の度合いについても扱っていなかった．

Adam らの形式化を基に，BDI エージェントと OCC theory を用いて感情を生起し，それに伴う行動を選択するエージェントの研究 [5][6] がある．[5] では，OCC theory によって分類された感情のうち，20 種類の感情が実装された．しかし，実装された感情には存在するか否かの判定しかなく，程度が無いため人間らしい感情表現として不十分であった．加えて，生起した感情は時間が経過しても消えることが無かった．

そこで本研究では，先行研究 [7] によって提案された論理体系を基に，程度の強さを取り入れた感情表現の実装を目指した．また，生起した感情を削除するシステムを導入することで，連続して感情を生起できるようにした．

2 OCC theory

OCC theory[3] とは，Ortony, Clore, Collins らが提唱した理論である．信念や願望などの心的状態を用いて人間の感情を 22 種類に分類，特徴付けている．感情の特徴付けが明確であり，計算機科学の分野において広く用いられている．

OCC theory によって分類された 22 種類の感情を図 1 に示す．なお，表における混合型とは，複数の感情が混合し生成した感情のことを指す．

<table>
<tr><td rowspan="4">1.
イベントの望ましさ</td><td colspan="4">（a）結果の望ましさに関して
（自分にとって）</td><td colspan="2">（b）結果の望ましさに関して
（他人にとって）</td></tr>
<tr><td>i.イベントに
関して</td><td colspan="3">ii.予想に関して</td><td>i.他人が良い結果を
得る</td><td>ii.他人が悪い結果を
得る</td></tr>
<tr><td rowspan="2">Joy
（喜び）
Distress
（嘆き）</td><td>A.単なる予想</td><td>B.予想したことが起こった</td><td>C.予想したことが
起こらなかった</td><td rowspan="2">HappyFor
（共に喜ぶ）
Resentment
（憤り）</td><td rowspan="2">SorryFor
（共に残念に思う）
Gloating
（ほくそ笑む）</td></tr>
<tr><td>Hope
（望み）
Fear
（恐れ）</td><td>Satisfaction
（満足）
FearConfirmed
（恐れていたことが確定）</td><td>Relief
（安堵）
Disappointment
（落胆）</td></tr>
<tr><td rowspan="2">2.
行動に対する
賞賛度に対して</td><td>a.自分の行動</td><td colspan="2">b.自分の行動とイベント
に関する混合型</td><td>c.他人の行動</td><td colspan="2">d.他人の行動とイベント
に関する混合型</td></tr>
<tr><td>Pride
（自尊心）
Shame
（羞恥心）</td><td colspan="2">Gratification
（満足）
Remorse
（後悔）</td><td>Admiration
（賞賛）
Reproach
（非難）</td><td colspan="2">Gratitude
（謝意）
Anger
（怒り）</td></tr>
<tr><td colspan="2">3.対象に対する好き嫌い</td><td colspan="5">Love（好き），Hate（嫌い）</td></tr>
</table>

図 1: OCC theory による分類

3 従来研究

従来研究 [5] では，Adam ら [1] の形式化を踏まえ 20 種類の感情を実装した．本章では，従来研究における実装について述べ，次いで従来研究において課題とされた点を述べる．

3.1 Adam らによる形式化

Adam らによる形式化では，感情の生起条件を BDI logic の論理式で表現している．図 1 における 22 種類の感情を以下の 7 グループに分類することで，グループ毎に同じ形式を持つ生起条件を用いることができる．

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
⑥ Well-being/Attribution compounds emotions：Gratification, Remorse, Gratitude, Anger
⑦ Attraction emotions：Love, Hate

Adam らによると，⑦の実装は，引数として対象物をとるため一階述語論理が必要である．しかし Adam らの論理体系では述語論理が扱えないため，形式化されていない．そのため，従来研究では⑦を除く 20 種類の感情表現の実装を行った．

次項から，生起条件の形式が同じものについてはグループの中から 1 つの感情を取り上げ，従来研究で実装された感情 (①〜⑥) の実装方法について説明する．

3.1.1 Well-being emotions

Joy（喜び）の生起条件は以下の通りである．

$$Joy_i\varphi \equiv Bel_i\varphi \wedge Des_i\varphi \tag{1}$$

これは「エージェント i がイベント 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 *Joy* という感情を生起する」となる．ここで，$Bel_i\varphi$ は「ト i は φ の成立を信じている」，$Des_i\varphi$ は「i にとって φ は望ましい」という意味である．

(1) 式をもとに，Jason で実装する．$Bel_i\varphi$ は，Jason 上の信念ベースで表現可能である．一方で，$Des_i\varphi$ は BDI モデルにおける Desire(願望) とは異なり，Desirability（i にとって φ は望ましい）を表している．そのため，Jason の願望にあたる Goal を用いて表現することができない．そこで，Adam らの形式化では $Des_i\varphi \Leftrightarrow Bel_iDes_i\varphi$ が成立していることを利用する．これは，$Des_i\varphi$ であることと i が $Des_i\varphi$ を信念に持つことが同等であることを指している．つまり，$Des_i\varphi$ を表現する場合，i の信念ベースに $Des_i\varphi$ を追加すれば良い．

よって，$Joy_i\varphi$ の実装は，信念ベースに φ と $Des_i\varphi$ の存在の有無を調べ，(1) 式の条件を満たせば $Joy_i\varphi$ を生起する，という方針をとっている．

3.1.2 Prospect emotions

Hope（望み）の生起条件は以下の通りである．

$$Hope_i\varphi \equiv Expect_i\varphi \wedge Des_i\varphi \tag{2}$$

これは「エージェント i はイベント 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 *Hope* という感情を生起する」となる．

$Expect_i\varphi$ は，$Prob_i\varphi \wedge \neg Bel_i\varphi$ と定義されており，$Prob_i\varphi$ は「φ である可能性が高い」という意味である．よって，$Expect_i\varphi$ は「φ を信じてはいないが，φ の発生はありえると思っている」という意味になる．

$Prob_i\varphi$ は確証度の低い信念とみなせるので，Jason では annotation 付き信念で表現できる．そのため，$Hope_i\varphi$ の実装は $Joy_i\varphi$ と同様の実装方針をとることができる．

3.1.3 Confirmation emotions

Satisfaction（満足）の生起条件は以下の通りである．

$$Satisfaction_i\varphi \equiv Bel_i\mathbf{P}Expect_i\varphi \wedge Des_i\varphi \wedge Bel_i\varphi \tag{3}$$

これは「エージェント i はイベント φ の成立を過去のある時点で期待し，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ば，i は φ を達成したことに対し *Satisfaction* という感情を生起する」となる．ここで，$\mathbf{P}$ は「過去のある時点」を表している．

しかし，過去の事象を全て記憶すると記憶が単調に増えつづけるため，$Bel_i\mathbf{P}Expect_i\varphi$ の実現は難しい．そこで，実装にあたっては (3) を以下のように書き換えている．

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

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」という意味になる．ここで，$\mathbf{A}(\psi\mathbf{N}\varphi)$ は「現在より後に，最初に φ が成立したとき ψ も成立する」という意味を指す．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．すなわち，$Des_i\varphi \Leftrightarrow \mathbf{G}Des_i\varphi$ が成り立つとみなすことができる．ここで，$\mathbf{G}$ は「未来永劫に」という意味を指しており，そのため，$Des_i\varphi$ を調べるタイミングはいつでも良いとされる．

3.1.4 Fortunes-of-others emotions

HappyFor（共に喜ぶ）の生起条件は以下の通りである．

$$HappyFor_{i,j}\varphi \equiv Bel_i\varphi \wedge Bel_iDes_j\varphi \wedge Des_iBel_j\varphi \tag{4}$$

これは「エージェント i にとって『エージェント j がイベント φ の成立を信じる』ことは望ましい ($Des_iBel_j\varphi$) かつ，i は『j にとって φ が望ましい』と信じる ($Bel_iDes_j\varphi$) かつ，i は φ の成立を信じる ($Bel_i\varphi$) ならば，i は φ の発生に対して，j のため *HappyFor* という感情を生起する」となる．

この感情を生起するため，他エージェントの心的状態を自エージェントが自身の信念として取得できるようにした．他エージェントの心的状態に依存する感情を実装するには，他エージェントの心的状態を調べる必要があった．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可能であり，他エージェントからはアクセスできない．そこで，他エージェントの信念が変わるたびに，その変更を自エージェントが認識できるよう実装を行った．複数のエージェント間の通信によって他エージェントの心的状態を自分の信念として取得し，これを用いて自身の感情を生起するよう実装している．

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利用する．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，その信念が他エージェントにも伝わるようにしている．

3.1.5 Attribution emotions

Pride（自尊心）の生起条件は以下の通りである．

$$Pride_i(i:\alpha,\varphi) \equiv Bel_iDone_{i:\alpha}(Idl_iHappens_{i:\alpha}\varphi \wedge Prob_iAfter_{i:\alpha}\neg\varphi) \wedge Bel_i\varphi \tag{5}$$

これは「『エージェント i がアクション α を実行して，イベント 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 $\neg\varphi$ が成立する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，$i$ は φ の成立を信じている ($Bel_i\varphi$) ならば，i は φ を達成したことに対して *Pride* という感情を生起する」となる．

要約すると,「達成が困難だと思われていたことを達成することで，i は φ の達成を誇りに思う」という意味である．

ここで，$Done_{i:\alpha}$ は「i が 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 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 α を実行すると φ が成立する」を意味する．

Adam らの形式化では，Idl は $Idl_iHappens_{i:\alpha}\varphi$ の形でのみ現れる．実装にあたり，これを i の信念ベースに $Idl\_Happens(\alpha,\varphi)$ の形で表現することで，$Idl_iHappens_{i:\alpha}\varphi$ と同等の情報を持つことができる．従って，これに加えて $Prob_iAfter_{i:\alpha}\neg\varphi$ が得られれば，$Idl_iHappens_{i:\alpha}\varphi \wedge Prob_iAfter_{i:\alpha}\neg\varphi$ が推論できる．よって，$Idl_iHappens_{i:\alpha}\varphi$ と $Prob_iAfter_{i:\alpha}\neg\varphi$ が信念として保持された後，さらに信念 φ が得られれば $Pride$ を生起するようにした．

また，任意のタイミングのイベントに対応するため，エージェントそれぞれに $Pride$ を生起する出来事のリストをあらかじめ用意しておく．任意のタイミングで追加された信念を，現在保持している信念とリストに一致するものはあるか調べ，一致するものがあれば $Pride$ を生起する．

なお，$Admiration$, $Reproach$ の他エージェントの行動評価に対する感情については，それぞれの生起条件を「j が φ を達成したとき」と定義することで実装している．

3.1.6 Well-being/Attribution compounds emotions

$Gratification$（満足）の生起条件は以下の通りである．

$$Gratification_i(i:\alpha,\varphi) \equiv Pride_i(i:\alpha,\varphi) \wedge Joy_i\varphi \tag{6}$$

Joy と $Pride$ の生起条件はそれぞれ 3.1.1 と 3.1.5 で述べてあるとおりであり，これは「エージェント i はイベント φ の達成に対して $Pride$ を生起し，かつ，φ の発生に対して Joy を生起するならば，$gratification$ という感情を生起する」という意味になる．

混合型の感情を実装するため，信念や願望がどのエージェントのものであるか明確にしている．感情の種類によっては，他エージェントとの通信を必要があり,「どのエージェントの持つ信念であるか」という情報を含む定義形式に統一することで感情生起のための信念や願望の追加が容易にした．

実装にあたり，$Pride$ を生起するための信念が，自分に取って望ましいものであるかを調べ，望ましければ Joy を生起する．これら 2 つの感情が成立したとき，$Gratification$ を生起するようにした．

3.2 実装上の問題点

従来研究では，一度生起した感情は時間が経過しても消えることが無く，また感情の度合いについて考慮されていなかった．また，信念の定義方法が複数あり，感情によっては生起する順番が変わると正しい感情を生起することができなかった．

次節以降で，本研究におけるこれらの改善について述べる．

4　感情の連続生起

感情を連続して生起するには，一度生起した感情を削除する必要がある．従来研究では，感情が生起された後に，その感情を操作することはできなかった．そこで，本研究では生起した感情を削除する機構を新たに導入した．本節ではその実装について述べる．

生起した感情を削除するには，外部から入力した信念や願望をすべて削除する必要がある．今回実装した 20 種類の感情が生起する場面は，以下の 2 通りに分類できる．外部からの入力をもとに感情を生起する場合と，外部からの入力によってエージェント内に派生した信念をもとに感情を生起する場合である．そこで，外部から入力された信念や願望，また派生した信念を取り除くことは，感情を削除するために有効な手段である．

実装方針として，あらかじめ外部からの入力を記憶しておき，感情の生起を確認した後に削除する.「all reset」と入力することで外部から入力された信念や願望を全て削除し,「エージェント名 reset」と入力することで，外部入力によってエージェント内に派生した信念を削除できるようにした．なお，本研究では外部からの入力によって派生する信念は Jason では全て prob という述語で表現されているため,「エージェント名 reset」と入力した際，そのエージェント内にある prob を全て削除するようにしている．信念を削除した後は，外部入力の記憶に用いていたファイルを削除する．

5　程度の強さを持つ感情表現

本研究では，Adam ら [1] の形式化を踏まえ，程度の強さを取り入れた論理体系 [7] をもとに実装を行った．本節ではその実装について述べる．程度の強さを取り入れるにあたり，時相オペレータや度合い付き信念などを Jason で利用可能な形で定義する．これを用い，生起条件を Jason の規則として記述することで，度合い付きの感情生起と，それに伴う行動決定を可能にした．また，感情の強さを計算する関数を新たに実装した．関数は，実装した 20 種類の感情それぞれに導入し，感情定義本体とは独立に変更可能としてある．

感情生起に関するプログラムは，Jason 上でエージェントを記述する asl ファイルとして実装した．

- em.asl：感情の生起条件
- con.asl：外部入力によって派生する信念やエージェント間の通信
- pride-shame.asl：Attribution emotions に必要なリスト作成や検査に必要なプラン
- act.asl：感情の度合いによって行動を選択するゴールに関するプラン
- degree.asl：感情の度合いを計算する関数

これらを各エージェントに include することで，感情生起を可能にした．

[7] での感情の定義には，*Bel*, *Des*, *Prob* などの様相オペレータを用いており，*Des*, *Prob* には度合いを表すパラメータがついている．本研究では，これらを Jason の述語で表し，度合いを表すパラメータは annotation で表現している．

$$Bel^i\varphi \equiv \mathrm{bel(X,Z)} \quad Des^i_d\varphi \equiv \mathrm{des(X,Z)[degOfCert(D)]} \quad Prob^i_l\varphi \equiv \mathrm{prob(X,Z)[degOfCert(D)]}$$

また，感情の種類や感情の程度の強さによって行動を選択する例を動作させることで，本研究の有用性を示した．本研究では各感情に対して，以下のように度合いによって行動を選択する例を設けている．

$D < 0.4$ のとき,「すこしの感情を生起したことを宣言する」行動を選択
$0.4 \leq D < 0.8$ のとき,「感情を生起したことを宣言する」行動を選択
$0.8 \leq D$ のとき,「とても感情を生起したことを宣言する」行動を選択

本研究でも [5] と同様に，同じ形式を持つ生起条件ごとに感情を分類，実装した．次項から各グループ毎に 1 つの感情を取り上げ，実装方法について説明する．なお，3.1 で述べた理由から⑥の実装は省いている．

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

5.1 Well-being emotions

程度の強さを持つ *Joy*（喜び）の生起条件は以下の通りである．

$$Joy^i_{f(d)}\varphi := Bel^i\varphi \wedge Des^i_d\varphi \tag{7}$$

これは「エージェント i がイベント φ の成立を信じ，かつそれが i にとって d 程度望ましいことであるならば，i は φ の発生に対して $f(d)$ 程度 *Joy* という感情を生起する」を意味する．

Joy の程度の強さに関係する変数 d として「i にとって φ がどの程度望ましいか $(Des^i_d\varphi)$」が導入されている．イベントの望ましさ d が大きいほど *Joy* の強さも大きくなると考えられているので，度合いを計算する $f(d)$ 関数は，望ましさ d を受け取り，d をそのまま *Joy* の強さとして返す恒等関数とした．

Jason で実装するにあたり，変数 JD, D を用いて，論理式の各項を以下のように表現した．また，$f(d)$ 関数は D を受け取って $f(D)$ を JD に代入する述語 joydeg(JD,D) で表現している．

$$Joy^i_{f(d)} \quad \equiv \quad \text{joy(X,Z)[degOfCert(JD)]}$$

JD：*Joy* の度合い　D：イベントの望ましさの度合い

エージェントは bel(X,Z) を得たときに *Joy* を生起する．実際の実装形式は図 2 のようになる．

イベントが起こるか否かについての可能性や，イベントを達成するため努力をしていないことを確認するため，not prob(X,Z), not prob(X,not Z), not effort(X,Z) を追加している．これは，*Hope* や *Satisfaction* 等の感情が生起した場合に *Joy* が重複して生起しないようにするためである．

5.2 Prospect emotions

程度の強さを持つ *Hope*（望み）の生起条件は以下の通りである．

$$Hope^i_{f(d,l)}\varphi := Des^i_d\varphi \wedge Prob^i_l\varphi \wedge l \neq 1 \tag{8}$$

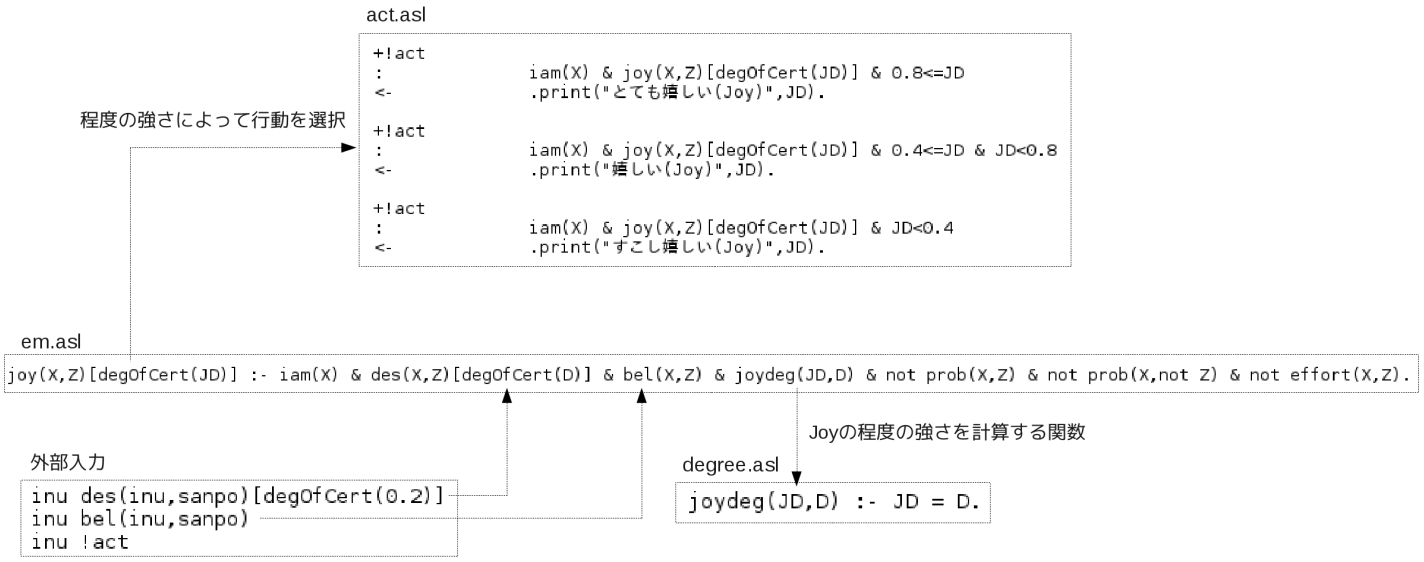

図 2: Joy

これは「エージェント i はイベント φ が l 程度成立しそうだと思っており（$l \neq 1$），かつそれが i にとって d 程度望ましいことであるならば，i は φ の発生に対して $f(d,l)$ 程度 *Hope* という感情を生起する」を意味する．

Hope の程度の強さに関係する変数として，*Joy* と同様に変数 d を利用する．それに加えて，新たな変数 l として「φ が起きている可能性 ($Prob_l^i\varphi$)」も導入されている．イベントの望ましさ d が大きいほど，また，イベントが起きている可能性 l が大きいほど *Hope* の強さも大きくなると考えられているので，度合いを計算する $f(d,l)$ 関数は，望ましさ d と可能性 l を加算した値を *Hope* の強さとして返す増加関数とした．

Jason で実装するにあたり，変数 HD, D1, D2 を用いて，論理式の各項を以下のように表現した．$f(d,l)$ 関数は，D1 と D2 を受け取って $f(D1,D2)$ を HD に代入する述語 hopedeg(HD,D1,D2) で表現している．

$$\begin{aligned} Hope^i_{f(d,l)} &\equiv \text{hope(X,Z)[degOfCert(HD)]} \\ Prob^i_l\varphi \wedge l \neq 1 &\equiv \text{expect(X,Z)[degOfCert(D2)]} \end{aligned}$$

HD：*Hope* の度合い　D1：イベントの望ましさの度合い　D2：イベントが起きている可能性

$Prob^i_l\varphi \wedge l \neq 1$ を表現するため，expect(X,Z)[degOfCert(D2)] という記法を用いている．イベントが起きている可能性 $l = 1$ のとき，イベントは実際に起きることを意味する．そこで，expect(X,Z) として，イベントが起きている可能性があることは信じている (prob(X,Z)[degOfCert(D2)]) が，実際にイベントが起きたという信念は持っていない (not bel(X,Z)) という意味を含むことで，$Prob^i_l\varphi \wedge l \neq 1$ の表現を可能にした．

エージェントは，prob(X,Z) を得て bel(X,Z) は得ていないとき，expect(X,Z) が成立し *Hope* を生起する．実際の実装形式は図 3 のようになる．

5.3 Prospect emotions

程度の強さを持つ *Satisfaction*（満足）の生起条件は以下の通りである．

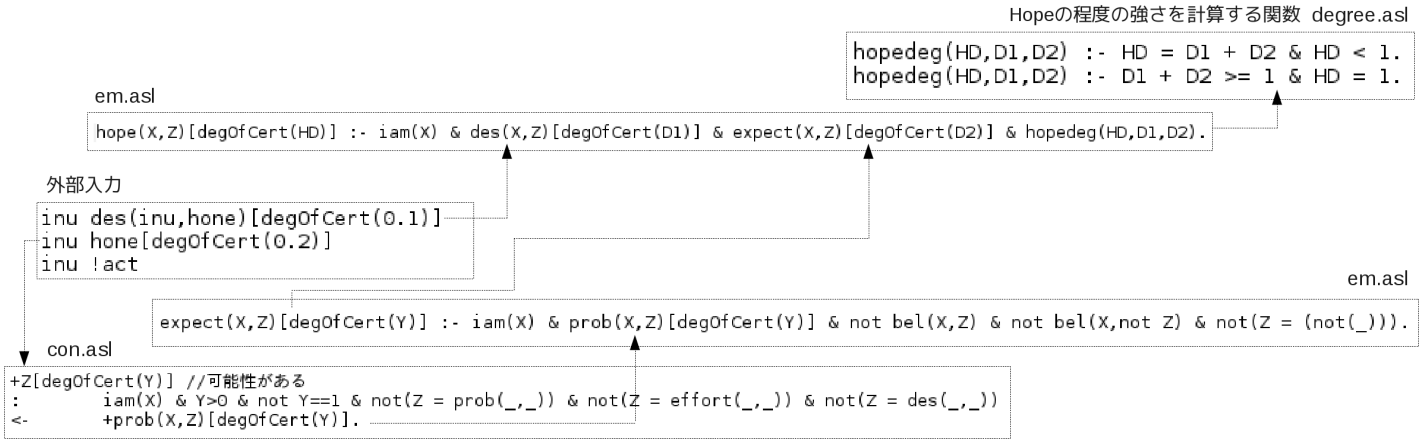

図 3: Hope

$$Satisfaction^i_{f(d,e,l)}\varphi := Des^i_d\varphi \wedge Bel^i\varphi \wedge Bel^i Past^i_e Prob^i_l\varphi \wedge l \neq 1 \tag{9}$$

これは「エージェント i はイベント φ が d 程度望ましく，かつ,『過去のある時点で φ の発生を l 程度期待し（$l \neq 1$），φ を達成するため e 程度努力をした』，かつ，現在は φ の成立を信じたならば，i は φ を達成したことに対し $f(d,e,l)$ 程度 *Satisfaction* という感情を生起する」を意味する．

Satisfaction の程度の強さに関係する変数として，これまでと同様に変数 d,l を利用する．それに加えて，新たな変数 e として「φ を達成するために費やした努力した程度」も導入されている．イベントの望ましさ d が大きいほど，また，イベントを達成するために費やした努力 e が大きいほど，そしてイベントが起きている可能性 l が大きいほど，*Satisfaction* の強さも大きくなると考えられているので，度合いを計算する $f(d,e,l)$ 関数は，望ましさ d と努力 e と可能性 l を加算した値を *Satisfaction* の強さとして返す増加関数とした．

Jason で実装するにあたり，変数 HD, D1, P, E, D2 を用いて，論理式の各項を以下のように表現した．$f(d,e,l)$ 関数は，hopedeg(HD,D1,D2) と同じ構造を持つ述語 ppdeg(D2,P,E) と satisdeg(SD,D1,D2) の 2 式で表現している．

$$Satisfaction^i_{f(d,e,l)} \equiv \text{satisfaction(X,Z)[degOfCert(SD)]}$$
$$Bel^i Past^i_e Prob^i_l\varphi \wedge l \neq 1 \equiv \text{past\_prob(X,Z)[degOfCert(D2)]}$$

SD：*Satisfaction* の度合い　D1：イベントの望ましさの度合い　P：イベントが起きている可能性
E：イベントを達成するため費やした努力の程度　D2：P と E を足し合わせた度合い

$Bel^i Past^i_e Prob^i_l\varphi \wedge l \neq 1$ を表現するため，past_prob(X,Z)[degOfCert(D2)] を用いている．まず，$Prob^i_l\varphi \wedge l \neq 1$ は *Hope* と同様に考え，prob(X,Z) はあるが bel(X,Z) はないと表現する．次に，$Past^i_e$ はイベントを達成するため努力した程度を扱う effort(X,Z)[degOfCert(E)] を新しく加えることで，$Bel^i Past^i_e Prob^i_l\varphi \wedge l \neq 1$ を表現した．

過去の時点で past_prob(X,Z) が成立し，現在 bel(X,Z) が成立するならば，エージェントは *Satisfaction* を生起する．実際の実装形式は図 4 のようになる．

5.4 Prospect emotions

程度の強さを持つ *HappyFor*（共に喜ぶ）の生起条件は以下の通りである．

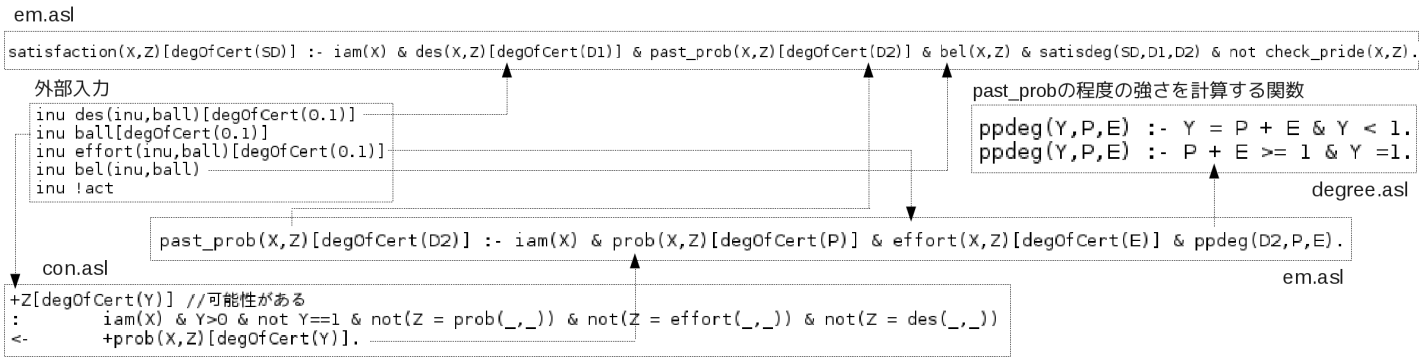

図 4: Satisfaction

$$HappyFor^{i,j}_{f(d_i,d_j)}\varphi := Bel^i\varphi \wedge Bel^i Des^j_{d_j}\varphi \wedge Des^i_{d_i} Bel^j\varphi \wedge Bel^i Deserve^j\varphi \tag{10}$$

これは「エージェント i は『エージェント j がイベント φ を d_j 程度望むこと』を信じており，かつ,『j が φ を達成すること』を d_i 程度望んでおり，かつ，i は φ が j に相応しいと思っているならば，i は j が φ を達成したことに対し $f(d_i, d_j)$ 程度 *HappyFor* という感情を生起する」を意味する．

HappyFor の程度の強さに関係する変数 d_i として「i にとって『j が φ を達成すること』がどの程度望ましいか $(Des^i_{d_i} Bel^j\varphi)$」と，変数 d_j として「j が φ をどの程度望ましいか $(Des^j_{d_j}\varphi)$」が定義されている．i と j のイベントに対する望ましさが大きいほど，*HappyFor* の強さも大きくなると考えられているので，度合いを計算する happyfordeg 関数は，望ましさ d_i と望ましさ d_j を加算した値を *HappyFor* の強さとして返している．

Jason で実装するにあたり，変数 HFD, D1, D2 を用いて，論理式の各項を以下のように表現した．$f(d_i, d_j)$ 関数は，前項と同様の構造を持つ述語 happyfordeg(HFD,D1,D2) で表現している．

$$\begin{aligned}
HappyFor^{i,j}_{f(d_i,d_j)} &\equiv \text{happyfor(X,Y,Z)[degOfCert(HFD)]} \\
Bel^i Des^j_{d_j}\varphi &\equiv \text{bel(X,des(Y,Z)[degOfCert(D1)])} \\
Des^i_{d_i} Bel^j\varphi &\equiv \text{des(X,bel(Y,Z))[degOfCert(D2)]} \\
Bel^i Deserve^j\varphi &\equiv \text{deserve(X,Y,Z)}
\end{aligned}$$

HFD：*HappyFor* の度合い　D1：Y が持つイベントの望ましさ　D2：X が持つイベントの望ましさ

$Bel^i Deserve^j\varphi$ を表現するにあたり，deserve(X,Y,Z) を用いている．これは，X は Y が Z を達成したいことを知っており (des(Y,Z))，かつ，X は Y が Z を達成することを望む (des(X,bel(Y,Z))) のであれば，すなわち X は Z が Y にとって相応しいと思うように定義している．

エージェントは他者との通信によって bel(X,des(Y,Z)) と bel(X,Z) を得て，deserve(X,Y,Z) が成立したときに *HappyFor* を生起する．実際の実装形式は図 5 のようになる．

5.5 Prospect emotions

程度の強さを持つ *Pride*（誇りに思う）の生起条件は以下の通りである．

$$Pride^i_{f(l,p)}\varphi := Bel^i PastProb^i_l\varphi \wedge Bel^i\varphi \wedge Praise^i_p\varphi \tag{11}$$

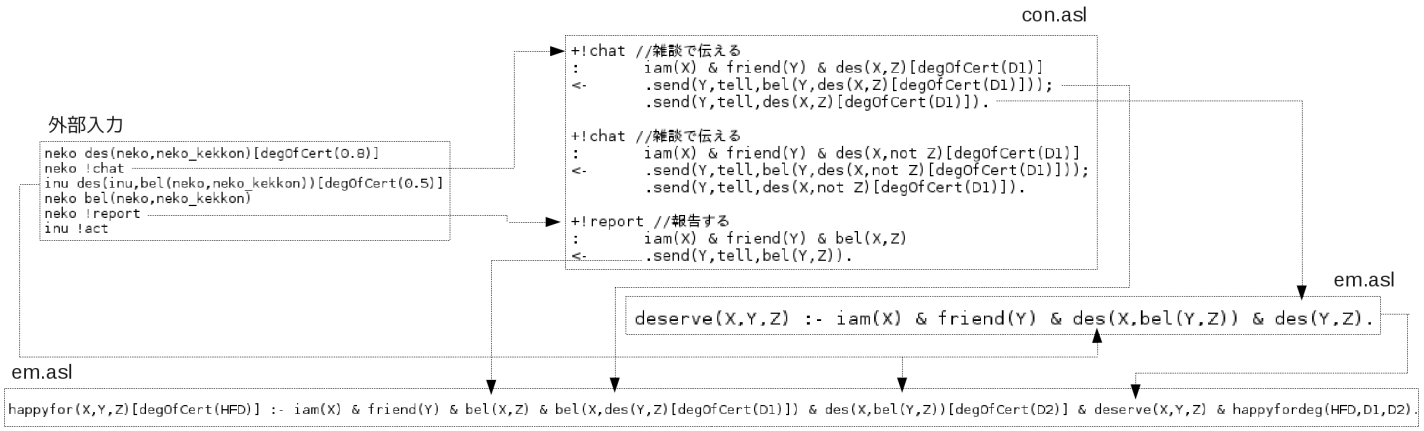

図 5: HappyFor

これは「エージェント i は過去の時点でイベント φ の可能性は l 程度低いと思っており，かつ，現在は φ を達成し，かつ，φ を達成したことは p 程度賞賛されると判断されるならば，i は φ を達成したことに対し $f(l,p)$ 程度 *Pride* という感情を生起する」を意味する．

Pride の程度の強さに関係する変数 l として，エージェントの行動の意外性 $(Bel^i\, PastProb_l^i\varphi \wedge Bel^i\varphi)$ が定義されている．この論理式は，過去に見込みが低いと思われていたことが，現在は真であることを意味している．また，変数 p としてイベントが賞賛に値すると判断される程度 $(Praise_p^i\varphi)$ が定義されている．行動の意外性 l が大きいほど，また，賞賛に値すると判断される程度 p が大きいほど，*Pride* の強さも大きくなると考えられているので，度合いを計算する prideg 関数は，意外性 l と賞賛度 p を加算した値を *Pride* の強さとして返している．

Jason で実装するにあたり，変数 PD, D1, D2, L を用いて，論理式の各項を以下のように表現した．行動の意外性 l は，イベントの見込みが低いほど，そのイベントを達成したときに値が大きくなると考えられる．そこで，prideg(PD,D1,D2) 関数に行動の意外性を表す変数 L= 1−D1 をおき，PD には L と D2 を加算した値を代入している．なお，イベントの見込みは低い必要があるので，D1 は 0.5 より小さい場合にのみ *Pride* を生起するようにした．

$$
\begin{aligned}
Pride_{f(l,p)}^i &\equiv \text{pride(X,Z)[degOfCert(PD)]} \\
Bel^i\, PastProb_l^i\varphi &\equiv \text{past\_prob(X,Z)[degOfCert(D1)]} \\
Praise_p^i\varphi &\equiv \text{praise(X,Z)[degOfCert(D2)]}
\end{aligned}
$$

PD：*Pride* の度合い　D1：イベントが起きている可能性　D2：賞賛に値すると判断される程度

$Praise_p^i\varphi$ を実装するにあたり，praise(X,Z)[degOfCert(D2)] を導入した．皆が達成したいと思うイベントを，エージェントが達成したとき賞賛されるに値すると思うようにする．そこで，エージェントが「Z は皆が D2 程度望むことである (des(everyone,Z)[degOfCert(D2)]」という信念を持つならば，そのイベントは賞賛に D2 程度値すると判断するようにした．

エージェントは prob(X,Z) と bel(X,Z) を得て，かつ past_prob(X,Z) と praise(X,Z) が成立したときに *Pride* を生起する．実際の実装形式は図 6 のようになる．

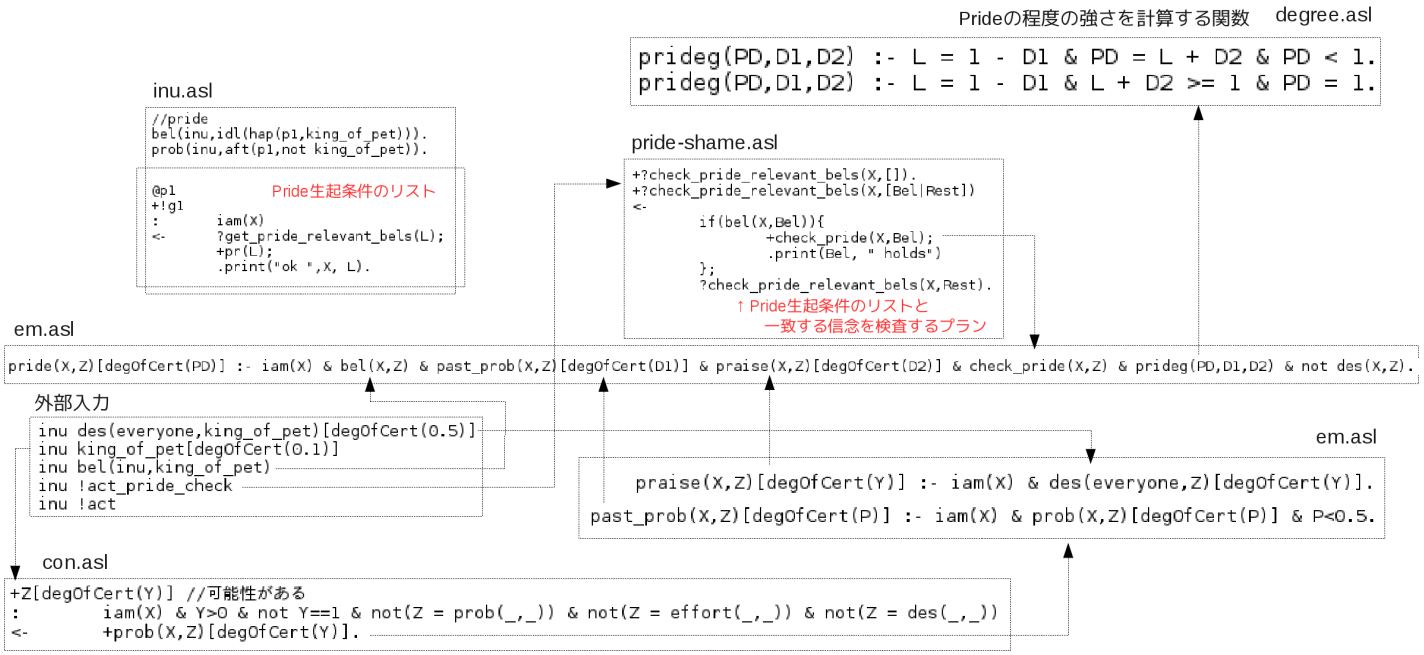

図 6: Pride

6　実験と結果

本章では，エージェントが感情を生起，行動決定するか検証する実験方法とその結果を述べる．

6.1　実験方法

実験は，生起した感情をもとに行動を決定するプランを Jason 上で記述し実行することで行った．各感情ごとに，あらかじめその感情を生起すると想定したシナリオを準備しておく．シナリオに従って想定した感情を生起し，生起した感情の種類や度合いによって正しい行動選択できることを確認する．

エージェントが感情を生起するにあたり，外部からエージェントの信念やゴールを変更をできるようにした．感情を生起するには，任意のタイミングで信念の追加や削除を行うイベントや，ゴールの追加を行うイベントを与える必要がある．そのため，外部から TCP の 49999 番ポートに接続し，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環境を用意した．

6.2　結果

6.2.1　*Joy*

Joy（喜び）を生起させるシナリオとして，以下のように設定する．

> エージェント inu がいるとする．inu は散歩がしたいと思っており，その願望の強さは D 程度である．実際に散歩することで inu の願望は達成され，inu は JD 程度喜ぶ．

図 7 を入力し，inu に感情を生起させる．想定通りの感情生起し，感情の度合いによって行動結果が変わることを確認した（図 8）．度合いが 0.4 より小さいとき「すこし嬉しい」，0.4 以上 0.8

より小さいとき「嬉しい」，0.8 以上のとき「とても嬉しい」と，5 章で設定したとおりの行動を選択している．

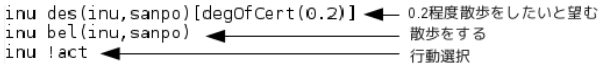

図 7: Joy を生起させる外部入力

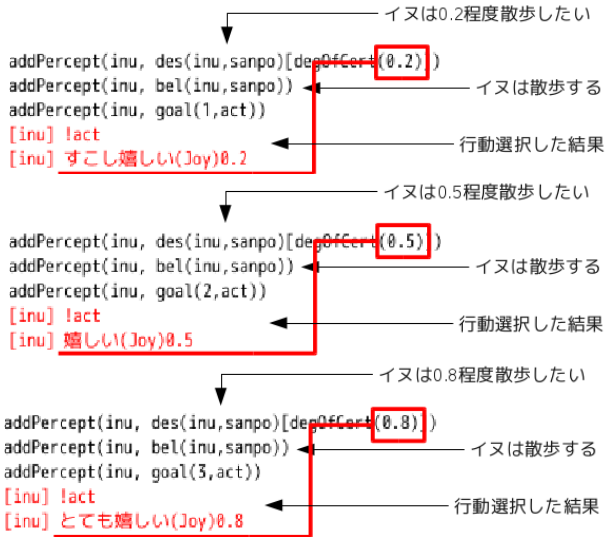

図 8: Joy を生起させた結果

6.2.2 *HappyFor*

HappyFor（共に喜ぶ）を生起させるシナリオとして，以下のように設定する．

> エージェント inu と neko がいるとする．neko は結婚したいと思っており，その願望の強さは D1 程度である．neko は，結婚したいことを inu に伝える．neko が結婚したいことを知った inu は，neko には結婚して欲しいと思い，その願望の強さは D2 程度である．neko は結婚し，そのことを inu に伝える．inu は neko が結婚したことを知り，HFD 程度共に喜ぶ．

図 9 を入力し，inu に感情を生起させる．想定通りの感情生起し，感情の度合いによって行動結果が変わることを確認した．また，ネコは自身の願望が叶ったので *Joy* という感情を別に生起していることを確認できた（図 10）．

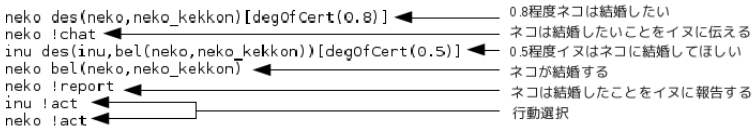

図 9: HappyFor を生起させる外部入力

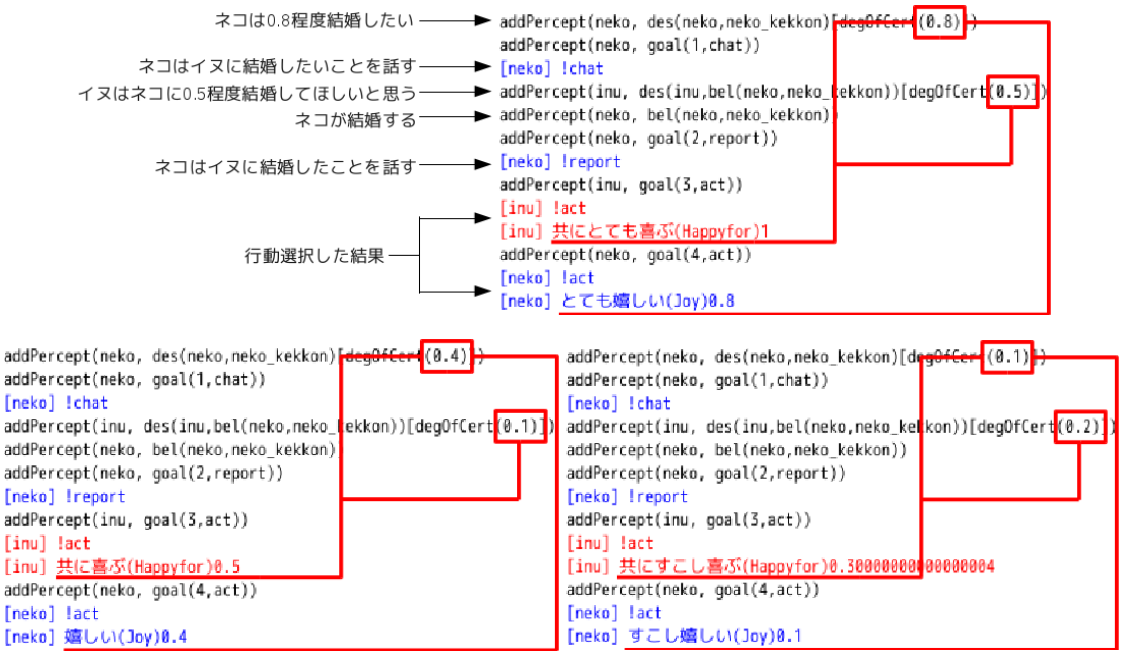

図 10: HappyFor を生起させた結果

6.2.3 感情の連続生起

感情を連続生起させるシナリオとして，以下のように設定する．

> エージェント inu と neko がいるとする．neko が離婚したことを聞き，inu は
> はじめほくそ笑む．しかし心を改め，さいごには共に悲しむ．

想定通りの感情生起し，感情の度合いによって行動結果が変わることを確認した（図 12）．

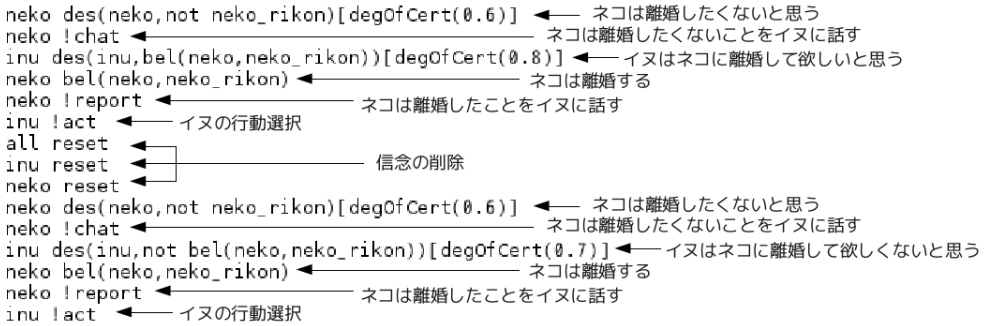

図 11: 感情の連続生起させる外部入力

7 終わりに

本研究では，より人間らしい感情表現の実現を目指し，OCC theory で扱われる 20 種類の感情に程度の強さを導入した．これにより，感情の種類のみならず，感情の度合いによる行動選択ができるようになった．

図 12: 感情の連続生起した結果

今後の課題として，生起した感情の妥当性を検討する必要がある．現段階では，感情の程度の強さを計算する関数は単純なものであり不十分である．また，1 つの感情を連続して生起することはできるが，同時に複数の感情を生起することはできない．複数の感情を同時に扱うとしても，特に，対立関係にある感情を同時に扱えるか否か考慮する必要がある．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できることを目指したい．

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から丁寧なご指導，ご助言を賜りました．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] Carole Adam, Andreas Herzig, and Dominique Longin. A logical formalization of OCC theory of emotions. *Synthese*, Vol. 168, No. 2, pp. 201–248, 2009.

[2] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming MultiAgent Systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.

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

[4] Anand S. Rao, Munindar P. Singh, and Michael P. Georgeff. *Formal Methods in DAI: Logic-Based Representation and Reasoning*. Massachusetts Institute of Technology, 1999.

[5] 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.

[6] 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.

[7] 池之内彰子, 新出尚之. OCC theory に基づくエージェントの感情表現の論理モデルについて. In *Proc. of JAWS2014*, 2014.

[8] 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.